MC68HC705P6A

Advance Information Data Sheet

M68HC05
Microcontrollers

MC68HC705P6A
Rev. 2.1
9/2005

freescale.com

This document contains certain information on a new product.Specifications and information herein are subject to change without notice.

MC68HC705P6A

Advance Information Data Sheet

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://www.freescale.com/

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Revision History

Date	Revision Level	Description	Page Number(s)
November, 2001	2.0	Format update to current publication standards	N/A
		Figure 11-1. Mask Option Register (MOR) — Definition of bit 6 corrected.	92
September, 2005	2.1	Updated to meet Freescale identity guidelines.	Throughout

List of Chapters

Table of Contents

Chapter 1
General Description

Chapter 2
Memory

Chapter 3
Operating Modes

Chapter 4
Resets

Chapter 5
Interrupts

Chapter 6
Input/Output Ports

Chapter 7
Serial Input/Output Port (SIOP)

Chapter 8
Capture/Compare Timer

Chapter 9
Analog Subsystem

Chapter 10
EPROM

Chapter 11
Mask Option Register (MOR)

Chapter 12
Central Processor Unit (CPU) Core

Chapter 13
Instruction Set

Chapter 14
Electrical Specifications

Chapter 15
Mechanical Specifications

Chapter 16
Ordering Information

Chapter 1
General Description

1.1 Introduction

The MC68HC705P6A is an EPROM version of the MC68HC05P6 microcontroller. It is a low-cost combination of an M68HC05 Family microprocessor with a 4-channel, 8-bit analog-to-digital (A/D) converter, a 16-bit timer with output compare and input capture, a serial communications port (SIOP), and a computer operating properly (COP) watchdog timer. The M68HC05 CPU core contains 176 bytes of RAM, 4672 bytes of user EPROM, 239 bytes of bootloader ROM, and 21 input/output (I/O) pins (20 bidirectional, 1 input-only). This device is available in either a 28-pin plastic dual in-line (PDIP) or a 28-pin small outline integrated circuit (SOIC) package.

A functional block diagram of the MC68HC705P6A is shown in Figure 1-1.

1.2 Features

Features of the MC68HC705P6A include:

- Low cost
- M68HC05 core
- 28-pin SOIC, PDIP, or windowed DIP package
- 4672 bytes of user EPROM (including 48 bytes of page zero EPROM and 16 bytes of user vectors)
- 239 bytes of bootloader ROM
- 176 bytes of on-chip RAM
- 4-channel 8-bit A/D converter
- SIOP serial communications port
- 16-bit timer with output compare and input capture
- 20 bidirectional I/O lines and 1 input-only line
- PC0 and PC1 high-current outputs
- Single-chip, bootloader, and test modes
- Power-saving stop, halt, and wait modes
- Static EPROM mask option register (MOR) selectable options:
  - COP watchdog timer enable or disable
  - Edge-sensitive or edge- and level-sensitive external interrupt
  - SIOP most significant bit (MSB) or least significant bit (LSB) first
  - SIOP clock rates: OSC divided by 8, 16, 32, or 64
  - Stop instruction mode, STOP or HALT
  - EPROM security external lockout
  - Programmable keyscan (pullups/interrupts) on PA0–PA7

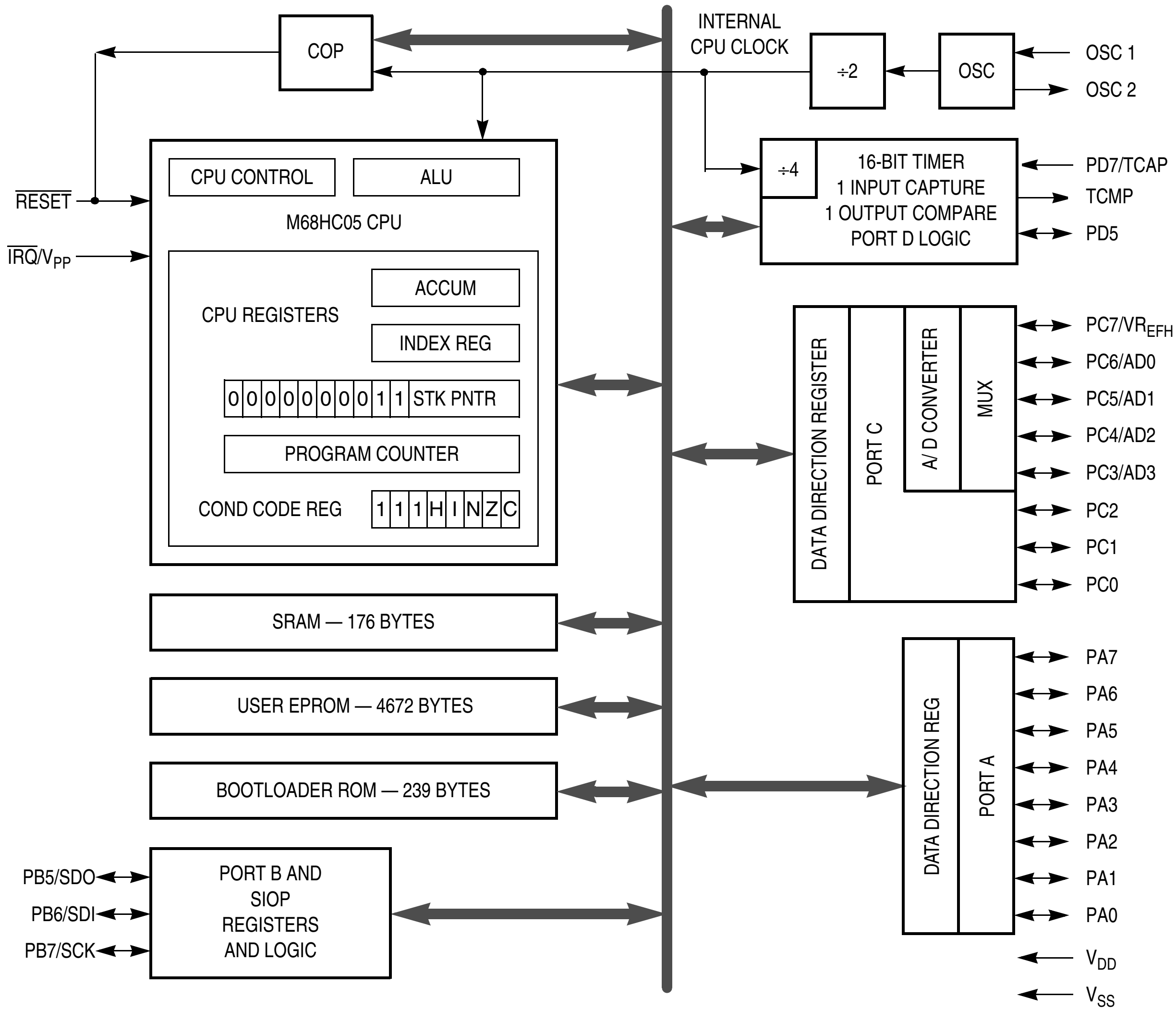


Figure 1-1. MC68HC705P6A Block Diagram

NOTE

A line over a signal name indicates an active low signal. For example, RESET is active high and RESET is active low.

Any reference to voltage, current, or frequency specified in the following sections will refer to the nominal values. The exact values and their tolerances or limits are specified in Chapter 14 Electrical Specifications.

1.3 Functional Pin Description

The following paragraphs describe the functionality of each pin on the MC68HC705P6A package. Pins connected to subsystems described in other chapters provide a reference to the chapter instead of a detailed functional description.

1.3.1 V_{DD} and V_{SS}

Power is supplied to the MCU through V_{DD} and V_{SS}. V_{DD} is connected to a regulated +5 volt supply and V_{SS} is connected to ground.

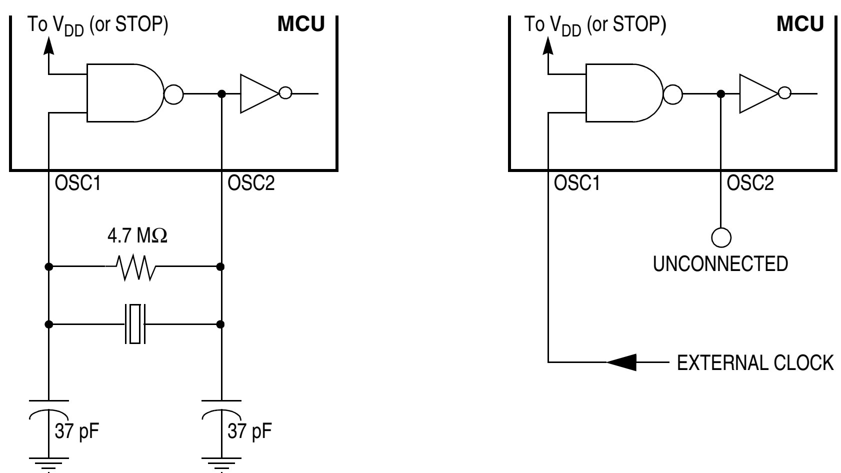
Very fast signal transitions occur on the MCU pins. The short rise and fall times place very high short-duration current demands on the power supply. To prevent noise problems, take special care to provide good power supply bypassing at the MCU. Use bypass capacitors with good high-frequency characteristics and position them as close to the MCU as possible. Bypassing requirements vary, depending on how heavily the MCU pins are loaded.

1.3.2 OSC1 and OSC2

The OSC1 and OSC2 pins are the control connections for the on-chip oscillator. The OSC1 and OSC2 pins can accept the following:

1. A crystal as shown in Figure 1-2(a)
2. A ceramic resonator as shown in Figure 1-2(a)
3. An external clock signal as shown in Figure 1-2(b)

The frequency, f_{osc}, of the oscillator or external clock source is divided by two to produce the internal bus clock operating frequency, f_{op}. The oscillator cannot be turned off by software unless the MOR bit, SWAIT, is clear when a STOP instruction is executed.



(a) Crystal or Ceramic Resonator Connections

(b) External Clock Source Connections

Figure 1-2. Oscillator Connections

1.3.2.1 Crystal

The circuit in Figure 1-2(a) shows a typical oscillator circuit for an AT-cut, parallel resonant crystal. Follow the crystal manufacturer's recommendations, as the crystal parameters determine the external component values required to provide maximum stability and reliable startup. The load capacitance values used in the oscillator circuit design should include all stray capacitances. Mount the crystal and components as close as possible to the pins for startup stabilization and to minimize output distortion.

1.3.2.2 Ceramic Resonator

In cost-sensitive applications, use a ceramic resonator in place of a crystal. Use the circuit in Figure 1-2(a) for a ceramic resonator and follow the resonator manufacturer's recommendations, as the resonator parameters determine the external component values required for maximum stability and reliable starting. The load capacitance values used in the oscillator circuit design should include all stray capacitances. Mount the resonator and components as close as possible to the pins for startup stabilization and to minimize output distortion.

1.3.2.3 External Clock

An external clock from another CMOS-compatible device can be connected to the OSC1 input, with the OSC2 input not connected, as shown in Figure 1-2(b).

1.3.3 $\overline{\text{RESET}}$

Driving this input low will reset the MCU to a known startup state. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger to improve its noise immunity. Refer to Chapter 4 Resets.

1.3.4 PA0–PA7

These eight I/O pins comprise port A. The state of any pin is software programmable and all port A lines are configured as inputs during power-on or reset. Port A has mask-option register enabled interrupt capability with internal pullup devices selectable for any pin. Refer to Chapter 6 Input/Output Ports.

1.3.5 PB5/SDO, PB6/SDI, and PB7/SCK

These three I/O pins comprise port B and are shared with the SIOP communications subsystem. The state of any pin is software programmable, and all port B lines are configured as inputs during power-on or reset. Refer to Chapter 6 Input/Output Ports and Chapter 7 Serial Input/Output Port (SIOP).

1.3.6 PC0-PC2, PC3/AD3, PC4/AD2, PC5/AD1, PC6/AD0, and PC7/V_{REFH}

These eight I/O pins comprise port C and are shared with the A/D converter subsystem. The state of any pin is software programmable and all port C lines are configured as inputs during power-on or reset. Refer to Chapter 6 Input/Output Ports and Chapter 9 Analog Subsystem.

1.3.7 PD5 and PD7/TCAP

These two I/O pins comprise port D and one of them is shared with the 16-bit timer subsystem. The state of PD5 is software programmable and is configured as an input during power-on or reset. PD7 is always an input. It may be read at any time, regardless of which mode of operation the 16-bit timer is in. Refer to Chapter 6 Input/Output Ports and Chapter 8 Capture/Compare Timer.

1.3.8 TCMP

This pin is the output from the 16-bit timer's output compare function. It is low after reset. Refer to Chapter 8 Capture/Compare Timer.

1.3.9 $\overline{IRQ}/V_{PP}$ (Maskable Interrupt Request)

This input pin drives the asynchronous interrupt function of the MCU in user mode and provides the V_{PP} programming voltage in bootloader mode. The MCU will complete the current instruction being executed before it responds to the $\overline{IRQ}$ interrupt request. When the $\overline{IRQ}/V_{PP}$ pin is driven low, the event is latched internally to signify an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch is set and the interrupt mask bit (I bit) in the condition code register is clear, the MCU will begin the interrupt sequence.

Depending on the MOR LEVEL bit, the $\overline{IRQ}/V_{PP}$ pin will trigger an interrupt on either a negative edge at the $\overline{IRQ}/V_{PP}$ pin and/or while the $\overline{IRQ}/V_{PP}$ pin is held in the low state. In either case, the $\overline{IRQ}/V_{PP}$ pin must be held low for at least one t_{ILIH} time period. If the edge- and level-sensitive mode is selected (LEVEL bit set), the $\overline{IRQ}/V_{PP}$ input pin requires an external resistor connected to V_{DD} for wired-OR operation. If the $\overline{IRQ}/V_{PP}$ pin is not used, it must be tied to the V_{DD} supply. The $\overline{IRQ}/V_{PP}$ pin input circuitry contains an internal Schmitt trigger to improve noise immunity. Refer to Chapter 5 Interrupts.

NOTE

If the voltage level applied to the $\overline{IRQ}/V_{PP}$ pin exceeds V_{DD}, it may affect the MCU's mode of operation. See Chapter 3 Operating Modes.

Chapter 2
Memory

2.1 Introduction

The MC68HC705P6A utilizes 13 address lines to access an internal memory space covering 8 Kbytes. This memory space is divided into I/O, RAM, ROM, and EPROM areas.

2.2 User Mode Memory Map

When the MC68HC705P6A is in the user mode, the 32 bytes of I/O, 176 bytes of RAM, 4608 bytes of user EPROM, 48 bytes of user page zero EPROM, 239 bytes of bootloader ROM, and 16 bytes of user vectors EPROM are all active as shown in Figure 2-1.

2.3 Bootloader Mode Memory Map

Memory space is identical to the user mode. See Figure 2-1.

2.4 Input/Output and Control Registers

Figure 2-2 and Figure 2-3 briefly describe the I/O and control registers at locations $0000–$001F. Reading unimplemented bits will return unknown states, and writing unimplemented bits will be ignored.

2.5 RAM

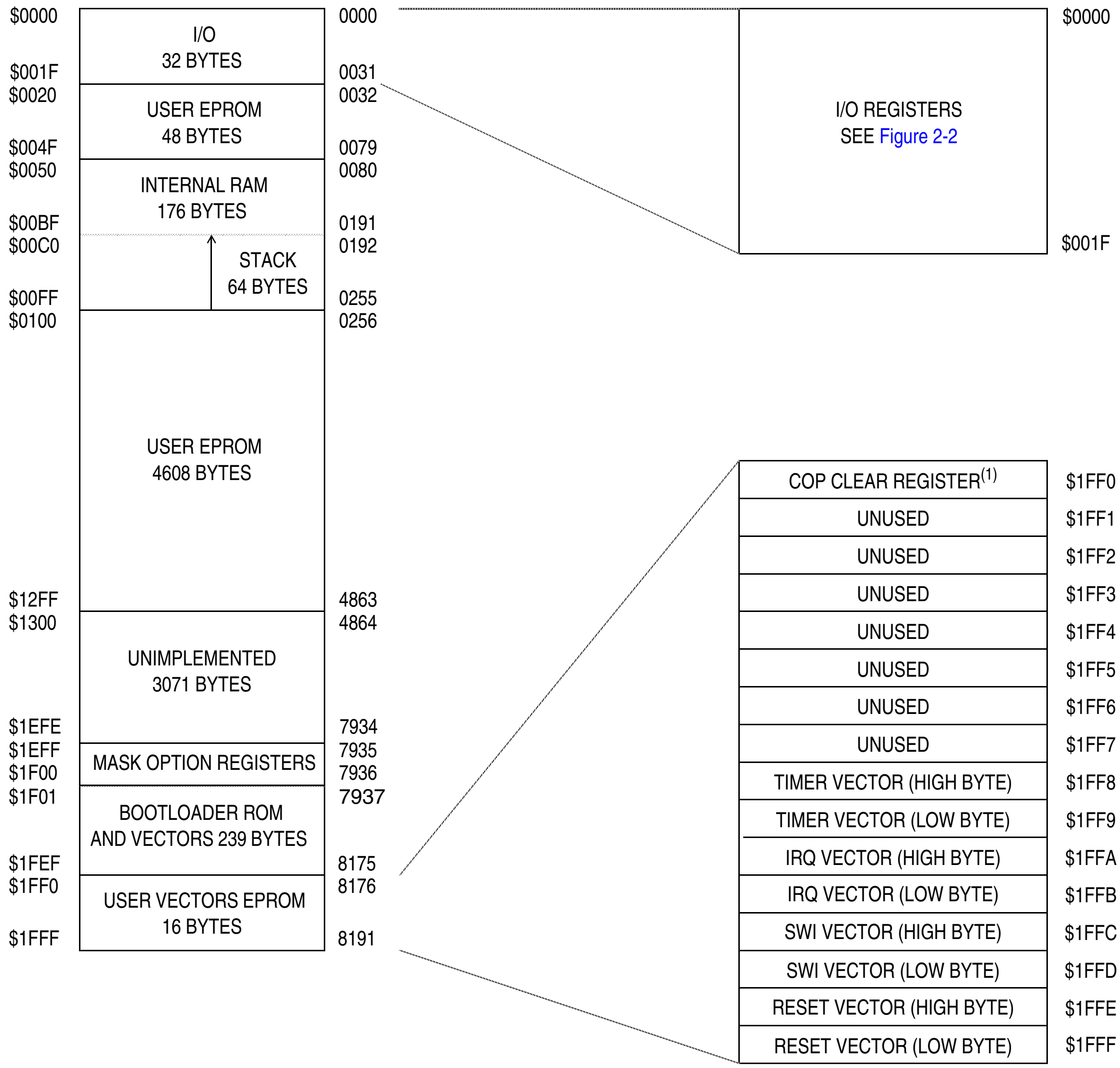
The user RAM consists of 176 bytes (including the stack) at locations $0050 through $00FF. The stack begins at address $00FF. The stack pointer can access 64 bytes of RAM from $00FF to $00C0.

NOTE

Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

2.6 EPROM/ROM

There are 4608 bytes of user EPROM at locations $0100 through $12FF, plus 48 bytes in user page zero locations $0020 through $004F, and 16 additional bytes for user vectors at locations $1FF0 through $1FFF. The bootloader ROM and vectors are at locations $1F01 through $1FEF.



Note 1. Writing zero to bit 0 of $1FF0 clears the COP watchdog timer. Reading $1FF0 returns user EPROM data.

Figure 2-1. MC68HC705P6A User Mode Memory Map

PORT A DATA REGISTER	$0000
PORT B DATA REGISTER	$0001
PORT C DATA REGISTER	$0002
PORT D DATA REGISTER	$0003
PORT A DATA DIRECTION REGISTER	$0004
PORT B DATA DIRECTION REGISTER	$0005
PORT C DATA DIRECTION REGISTER	$0006
PORT D DATA DIRECTION REGISTER	$0007
UNIMPLEMENTED	$0008
UMIMPLEMENTED	$0009
SIOP CONTROL REGISTER	$000A
SIOP STATUS REGISTER	$000B
SIOP DATA REGISTER	$000C
RESERVED	$000D
UNIMPLEMENTED	$000E
UNIMPLEMENTED	$000F
UNIMPLEMENTED	$0010
UNIMPLEMENTED	$0011
TIMER CONTROL REGISTER	$0012
TIMER STATUS REGISTER	$0013
INPUT CAPTURE MSB	$0015
INPUT CAPTURE LSB	$0016
OUTPUT COMPARE MSB	$0017
OUTPUT COMPARE LSB	$0017
TIMER MSB	$0018
TIMER LSB	$0019
ALTERNATE COUNTER MSB	$001A
ALTERNATE COUNTER LSB	$001B
EPROM PROGRAMMING REGISTER	$001C
A/D CONVERTER DATA REGISTER	$001D
A/D CONVERTER CONTROL AND STATUS REGISTER	$001E
RESERVED	$001F

Figure 2-2. MC68HC705P6A I/O and Control Registers Memory Map

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PORTA) See page 37.	Read:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		Write:								
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PORTB) See page 38.	Read:	PB7	PB6	PB5	0	0	0	0	0
		Write:								
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PORTC) See page 38.	Read:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		Write:								
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PORTD) See page 39.	Read:	PD7	0	PD5	1	0	0	0	0
		Write:								
		Reset:	Unaffected by reset							
$0004	Port A Data Direction Register (DDRA) See page 37.	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0005	Port B Data Direction Register (DDRB) See page 38.	Read:	DDRB7	DDRB6	DDRB5	1	1	1	1	1
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0006	Port C Data Direction Register (DDRC) See page 38.	Read:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007	Port D Data Direction Register (DDRD) See page 39.	Read:	0	0	DDRD5	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0008	Unimplemented									
$0009	Unimplemented									
$000A	SIOP Control Register (SCR) See page 43.	Read:	0	SPE	0	MSTR	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000B	SIOP Status Register (SSR) See page 44.	Read:	SPIF	DCOL	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000C	SIOP Data Register (SDR) See page 44.	Read:	SDR7	SDR6	SDR5	SDR4	SDR3	SSDR2	SDR1	SDR0
		Write:								
		Reset:	Unaffected by reset							

(shaded) = Unimplemented   R = Reserved   U = Undetermined

Figure 2-3. I/O and Control Register Summary (Sheet 1 of 3)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000D	Reserved for Test		R	R	R	R	R	R	R	R
$000E	Unimplemented									
$000F	Unimplemented									
$0010	Unimplemented									
$0011	Unimplemented									
$0012	Timer Control Register (TCR) See page 47.	Read:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
		Write:								
		Reset:	0	0	0	0	0	0	U	0
$0013	Timer Status Register (TSR) See page 48.	Read:	ICF	OCF	TOF	0	0	0	0	0
		Write:								
		Reset:	U	U	U	0	0	0	0	0
$0014	Input Capture Register MSB (ICRH) See page 50.	Read:	ICRH7	ICRH6	ICRH5	ICRH4	ICRH3	ICRH2	ICRH1	ICRH0
		Write:								
		Reset:	Unaffected by reset							
$0015	Input Capture Register LSB (ICRL) See page 50.	Read:	ICRL7	ICRL6	ICRL5	ICRL4	ICRL3	ICRL2	ICRL1	ICRL0
		Write:								
		Reset:	Unaffected by reset							
$0016	Output Compare Register MSB (OCRH) See page 50.	Read:	OCRH7	OCRH6	OCRH5	OCRH4	OCRH3	OCRH2	OCRH1	OCRH0
		Write:								
		Reset:	Unaffected by reset							
$0017	Output Compare Register LSB (OCRL) See page 50.	Read:	OCRL7	OCRL6	OCRL5	OCRL4	OCRL3	OCRL2	OCRL1	OCRL0
		Write:								
		Reset:	Unaffected by reset							
$0018	Timer Register MSB (TRH) See page 49.	Read:	TRH7	TRH6	TRH5	TRH4	TRH3	TRH2	TRH1	TRH0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0019	Timer Register LSB (TRL) See page 49.	Read:	TRL7	TRL6	TRL5	TRL4	TRL3	TRL2	TRL1	TRL0
		Write:								
		Reset:	1	1	1	1	1	1	0	0
$001A	Alternate Timer Register MSB (ATRH) See page 49.	Read:	ACRH7	ACRH6	ACRH5	ACRH4	ACRH3	ACRH2	ACRH1	ACRH0
		Write:								
		Reset:	1	1	1	1	1	1	1	1

[shaded] = Unimplemented R = Reserved U = Undetermined

Figure 2-3. I/O and Control Register Summary (Sheet 2 of 3)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001B	Alternate Timer Register LSB (ATRL)	Read:	ACRL7	ACRL6	ACRL5	ACRL4	ACRL3	ACRL2	ACRL1	ACRL0
		Write:								
	See page 49.	Reset:	1	1	1	1	1	1	0	0
$001C	EPROM Programming Register (EPROG)	Read:	0	0	0	0	0	ELAT	0	EPGM
		Write:								
	See page 58.	Reset:	0	0	0	0	0	0	0	0
$001D	A/D Conversion Value Data Register (ADC)	Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		Write:								
	See page 55.	Reset:	Unaffected by reset							
$001E	A/D Status and Control Register (ADSC)	Read:	CC	ADRC	ADON	0	0	CH2	CH1	CH0
		Write:								
	See page 54.	Reset:	0	0	0	0	0	0	0	0
$001F	Reserved for Test		R	R	R	R	R	R	R	R

[shaded] = Unimplemented R = Reserved U = Undetermined

Figure 2-3. I/O and Control Register Summary (Sheet 3 of 3)

2.7 Mask Option Register

The mask option register (MOR) is a pair of EPROM bytes located at $1EFF and $1F00. It controls the programmable options on the MC68HC705P6A. See Chapter 11 Mask Option Register (MOR) for additional information.

$1EFF	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PA7PU	PA6PU	PA5PU	PA4PU	PA3PU	PA2PU	PA1PU	PA0PU
Erased State:	0	0	0	0	0	0	0	0

$1F00	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SECURE		SWAIT	SPR1	SPR0	LSBF	LEVEL	COP
Erased State:	0	0	0	0	0	0	0	0

[shaded] = Unimplemented

Figure 2-4. Mask Option Register (MOR)

2.8 Computer Operating Properly (COP) Clear Register

The computer operating properly (COP) watchdog timer is located at address $1FF0. Writing a logical 0 to bit zero of this location will clear the COP watchdog counter as described in 4.3.2 Computer Operating Properly (COP) Reset.

$1FF0	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	0
Write:								COPR
Reset:	0	0	0	0	0	0	0	0

(shaded) = Unimplemented

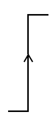
Figure 2-5. COP Watchdog Timer Location

Chapter 3
Operating Modes

3.1 Introduction

The MC68HC705P6A has two modes of operation that affect the pinout and architecture of the MCU: user mode and bootloader mode. The user mode is normally used for the application and the bootloader mode is used for programming the EPROM. The conditions required to enter each mode are shown in Table 3-1. The mode of operation is determined by the voltages on the $\overline{\mathrm{IRQ}}/V_{PP}$ and PD7/TCAP pins on the rising edge of the external $\overline{\mathrm{RESET}}$ pin.

Table 3-1. Operating Mode Conditions After Reset

$\overline{\mathrm{RESET}}$ Pin	$\overline{\mathrm{IRQ}}/V_{PP}$	PD7/TCAP	Mode
	V_{SS} to V_{DD}	V_{SS} to V_{DD}	Single chip
	V_{PP}	V_{DD}	Bootloader

The mode of operation is also determined whenever the internal computer operating properly (COP) watchdog timer resets the MCU. When the COP timer expires, the voltage applied to the $\overline{\mathrm{IRQ}}/V_{PP}$ pin controls the mode of operation while the voltage applied to PD7/TCAP is ignored. The voltage applied to PD7/TCAP during the last rising edge on $\overline{\mathrm{RESET}}$ is stored in a latch and used to determine the mode of operation when the COP watchdog timer resets the MCU.

3.2 User Mode

The user mode allows the MCU to function as a self-contained microcontroller, with maximum use of the pins for on-chip peripheral functions. All address and data activity occurs within the MCU and are not available externally. User mode is entered on the rising edge of $\overline{\mathrm{RESET}}$ if the $\overline{\mathrm{IRQ}}/V_{PP}$ pin is within the normal operating voltage range. The pinout for the user mode is shown in Figure 3-1.

In the user mode, there is an 8-bit I/O port, a second 8-bit I/O port shared with the analog-to-digital (A/D) subsystem, one 3-bit I/O port shared with the serial input/output port (SIOP), and a 3-bit port shared with the 16-bit timer subsystem, which includes one general-purpose I/O pin.

3.3 Bootloader Mode

The bootloader mode provides a means to program the user EPROM from an external memory device or host computer. This mode is entered on the rising edge of $\overline{\mathrm{RESET}}$ if V_{PP} is applied to the $\overline{\mathrm{IRQ}}/V_{PP}$ pin and V_{DD} is applied to the PD7/TCAP pin. The user code in the external memory device must have data located in the same address space it will occupy in the internal MCU EPROM, including the mask option register (MOR) at \$1EFF and \$1F00.

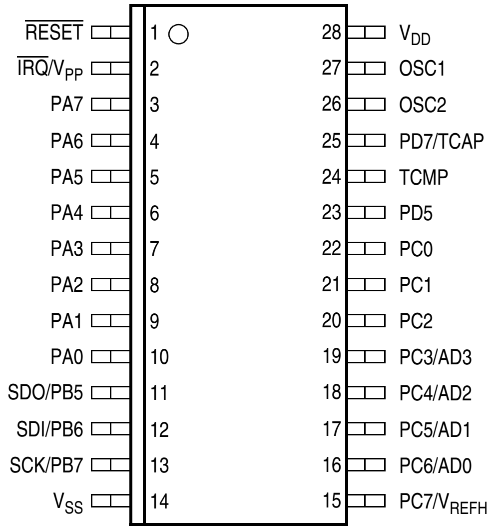


Figure 3-1. User Mode Pinout

3.4 Low-Power Modes

The MC68HC705P6A is capable of running in a low-power mode in each of its configurations. The WAIT and STOP instructions provide three modes that reduce the power required for the MCU by stopping various internal clocks and/or the on-chip oscillator. The SWAIT bit in the MOR is used to modify the behavior of the STOP instruction from stop mode to halt mode. The flow of the stop, halt, and wait modes is shown in Figure 3-2.

3.4.1 STOP Instruction

The STOP instruction can result in one of two modes of operation depending on the state of the SWAIT bit in the MOR. If the SWAIT bit is clear, the STOP instruction will behave like a normal STOP instruction in the M68HC05 Family and place the MCU in stop mode. If the SWAIT bit in the MOR is set, the STOP instruction will behave like a WAIT instruction (with the exception of a brief delay at startup) and place the MCU in halt mode.

3.4.1.1 Stop Mode

Execution of the STOP instruction when the SWAIT bit in the MOR is clear places the MCU in its lowest power consumption mode. In stop mode, the internal oscillator is turned off, halting *all* internal processing, including the COP watchdog timer. Execution of the STOP instruction automatically clears the I bit in the condition code register so that the $\overline{\text{IRQ}}$ external interrupt is enabled. All other registers and memory remain unaltered. All input/output lines remain unchanged.

The MCU can be brought out of stop mode only by an $\overline{\text{IRQ}}$ external interrupt or an externally generated $\overline{\text{RESET}}$. When exiting stop mode, the internal oscillator will resume after a 4064 internal clock cycle oscillator stabilization delay.

NOTE

Execution of the STOP instruction when the SWAIT bit in the MOR is clear will cause the oscillator to stop, and, therefore, disable the COP watchdog timer. To avoid turning off the COP watchdog timer, stop mode should be changed to halt mode by setting the SWAIT bit in the MOR. See 3.5 COP Watchdog Timer Considerations for additional information.

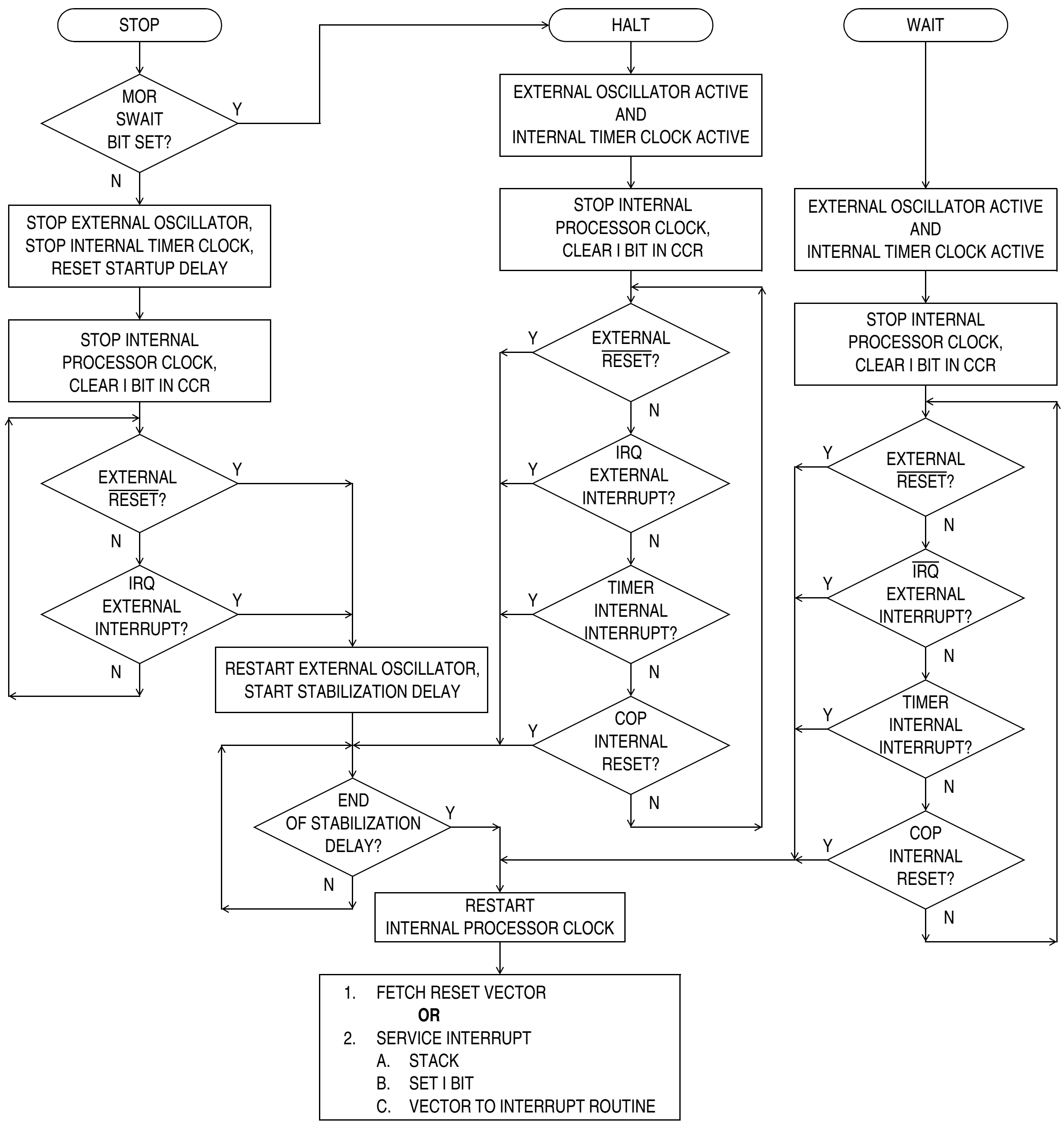


Figure 3-2. STOP/WAIT Flowcharts

3.4.1.2 Halt Mode

NOTE

Halt mode is ***NOT*** *designed for intentional use. Halt mode is only provided to keep the COP watchdog timer active in the event a STOP instruction is executed inadvertently. This mode of operation is usually achieved by invoking wait mode.*

Execution of the STOP instruction when the SWAIT bit in the MOR is set places the MCU in this low-power mode. Halt mode consumes the same amount of power as wait mode (both halt and wait modes consume more power than stop mode).

In halt mode, the internal clock is halted, suspending all processor and internal bus activity. Internal timer clocks remain active, permitting interrupts to be generated from the 16-bit timer or a reset to be generated from the COP watchdog timer. Execution of the STOP instruction automatically clears the I bit in the condition code register, enabling the $\overline{\text{IRQ}}$ external interrupt. All other registers, memory, and input/output lines remain in their previous states.

If the 16-bit timer interrupt is enabled, it will cause the processor to exit the halt mode and resume normal operation. The halt mode also can be exited when an $\overline{\text{IRQ}}$ external interrupt or external $\overline{\text{RESET}}$ occurs. When exiting the halt mode, the internal clock will resume after a delay of one to 4064 internal clock cycles. This varied delay time is the result of the halt mode exit circuitry testing the oscillator stabilization delay timer (a feature of the stop mode), which has been free-running (a feature of the wait mode).

3.4.2 WAIT Instruction

The WAIT instruction places the MCU in a low-power mode which consumes more power than stop mode. In wait mode, the internal clock is halted, suspending all processor and internal bus activity. Internal timer clocks remain active, permitting interrupts to be generated from the 16-bit timer and reset to be generated from the COP watchdog timer. Execution of the WAIT instruction automatically clears the I bit in the condition code register, enabling the $\overline{\text{IRQ}}$ external interrupt. All other registers, memory, and input/output lines remain in their previous state.

If the 16-bit timer interrupt is enabled, it will cause the processor to exit wait mode and resume normal operation. The 16-bit timer may be used to generate a periodic exit from wait mode. Wait mode may also be exited when an $\overline{\text{IRQ}}$ external interrupt or $\overline{\text{RESET}}$ occurs.

3.5 COP Watchdog Timer Considerations

The COP watchdog timer is active in user mode of operation when the COP bit in the MOR is set. Executing the STOP instruction when the SWAIT bit in the MOR is clear will cause the COP to be disabled. Therefore, it is recommended that the STOP instruction be modified to produce halt mode (set bit SWAIT in the MOR) if the COP watchdog timer is required to function at all times.

Furthermore, it is recommended that the COP watchdog timer be disabled for applications that will use the wait mode for time periods that will exceed the COP timeout period.

Chapter 4
Resets

4.1 Introduction

The MCU can be reset from three sources: one external input and two internal reset conditions. The $\overline{RESET}$ pin is a Schmitt trigger input as shown in Figure 4-1. The CPU and all peripheral modules will be reset by the RST signal which is the logical OR of internal reset functions and is clocked by PH1.

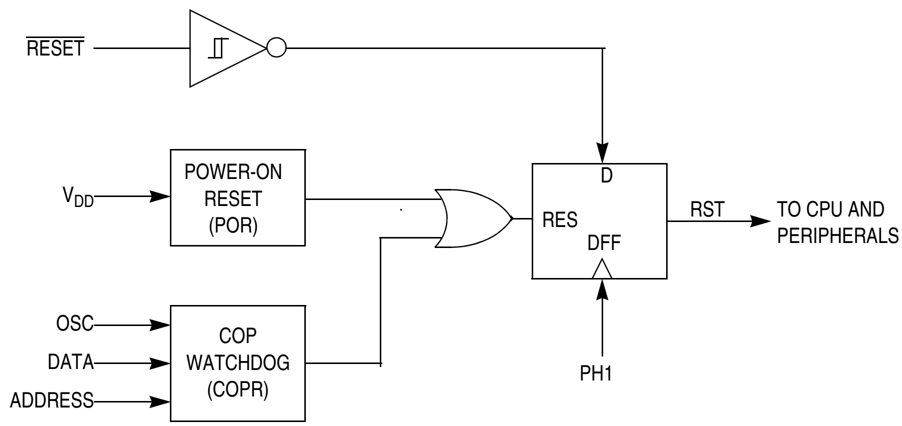


Figure 4-1. Reset Block Diagram

4.2 External Reset ($\overline{RESET}$)

The $\overline{RESET}$ input is the only external reset and is connected to an internal Schmitt trigger. The external reset occurs whenever the $\overline{RESET}$ input is driven below the lower threshold and remains in reset until the $\overline{RESET}$ pin rises above the upper threshold. The upper and lower thresholds are given in Chapter 14 Electrical Specifications.

4.3 Internal Resets

The two internally generated resets are the initial power-on reset (POR) function and the computer operating properly (COP) watchdog timer function.

4.3.1 Power-On Reset (POR)

The internal POR is generated at power-up to allow the clock oscillator to stabilize. The POR is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal clock cycle oscillator stabilization delay after the oscillator becomes active.

The POR will generate the RST signal and reset the MCU. If any other reset function is active at the end of this 4064 internal clock cycle delay, the RST signal will remain active until the other reset condition(s) end.

4.3.2 Computer Operating Properly (COP) Reset

When the COP watchdog timer is enabled (COP bit in the MOR is set), the internal COP reset is generated automatically by a timeout of the COP watchdog timer. This timer is implemented with an 18-stage ripple counter that provides a timeout period of 65.5 ms when a 4-MHz oscillator is used. The COP watchdog counter is cleared by writing a logical 0 to bit zero at location $1FF0.

The COP watchdog timer can be disabled by clearing the COP bit in the MOR or by applying 2 x V_{DD} to the $\overline{IRQ}/V_{PP}$ pin (for example, during bootloader). When the $\overline{IRQ}/V_{PP}$ pin is returned to its normal operating voltage range (between V_{SS}–V_{DD}), the COP watchdog timer's output will be restored if the COP bit in the mask option register (MOR) is set.

The COP register is shared with the least significant byte (LSB) of an unused vector address as shown in Figure 4-2. Reading this location will return the programmed value of the unused user interrupt vector, usually 0. Writing to this location will clear the COP watchdog timer.

Address: $1FF0

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	0
Write:								COPR

(shaded) = Unimplemented

Figure 4-2. Unused Vector and COP Watchdog Timer

When the COP watchdog timer expires, it will generate the RST signal and reset the MCU. If any other reset function is active at the end of the COP reset signal, the RST signal will remain in the reset condition until the other reset condition(s) end. When the reset condition ends, the MCU's operating mode will be selected (see Table 3-1. Operating Mode Conditions After Reset).

Chapter 5
Interrupts

5.1 Introduction

The MCU can be interrupted six different ways:

1. Non-maskable software interrupt instruction (SWI)
2. External asynchronous interrupt ($\overline{IRQ}$)
3. Input capture interrupt (TIMER)
4. Output compare interrupt (TIMER)
5. Timer overflow interrupt (TIMER)
6. Port A interrupt (if selected via mask option register)

Interrupts cause the processor to save the register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. Unlike reset, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is completed.

When the current instruction is completed, the processor checks all pending hardware interrupts. If interrupts are not masked (I bit in the condition code register is clear) and the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing. Otherwise, the next instruction is fetched and executed. The SWI is executed the same as any other instruction, regardless of the I-bit state.

When an interrupt is to be processed, the CPU puts the register contents on the stack, sets the I bit in the CCR, and fetches the address of the corresponding interrupt service routine from the vector table at locations $1FF8 through $1FFF. If more than one interrupt is pending when the interrupt vector is fetched, the interrupt with the highest vector location shown in Table 5-1 will be serviced first.

Table 5-1. Vector Addresses for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	Reset	RESET	$1FFE–$1FFF
N/A	N/A	Software	SWI	$1FFC–$1FFD
N/A	N/A	External Interrupt	IRQ	$1FFA–$1FFB
TSR	ICF	Timer Input Capture	TIMER	$1FF8–$1FF9
TSR	OCF	Timer Output Compare	TIMER	$1FF8–$1FF9
TSR	TOF	Timer Overflow	TIMER	$1FF8–$1FF9

An RTI instruction is used to signify when the interrupt software service routine is completed. The RTI instruction causes the CPU state to be recovered from the stack and normal processing to resume at the next instruction that was to be executed when the interrupt took place. Figure 5-1 shows the sequence of events that occurs during interrupt processing.

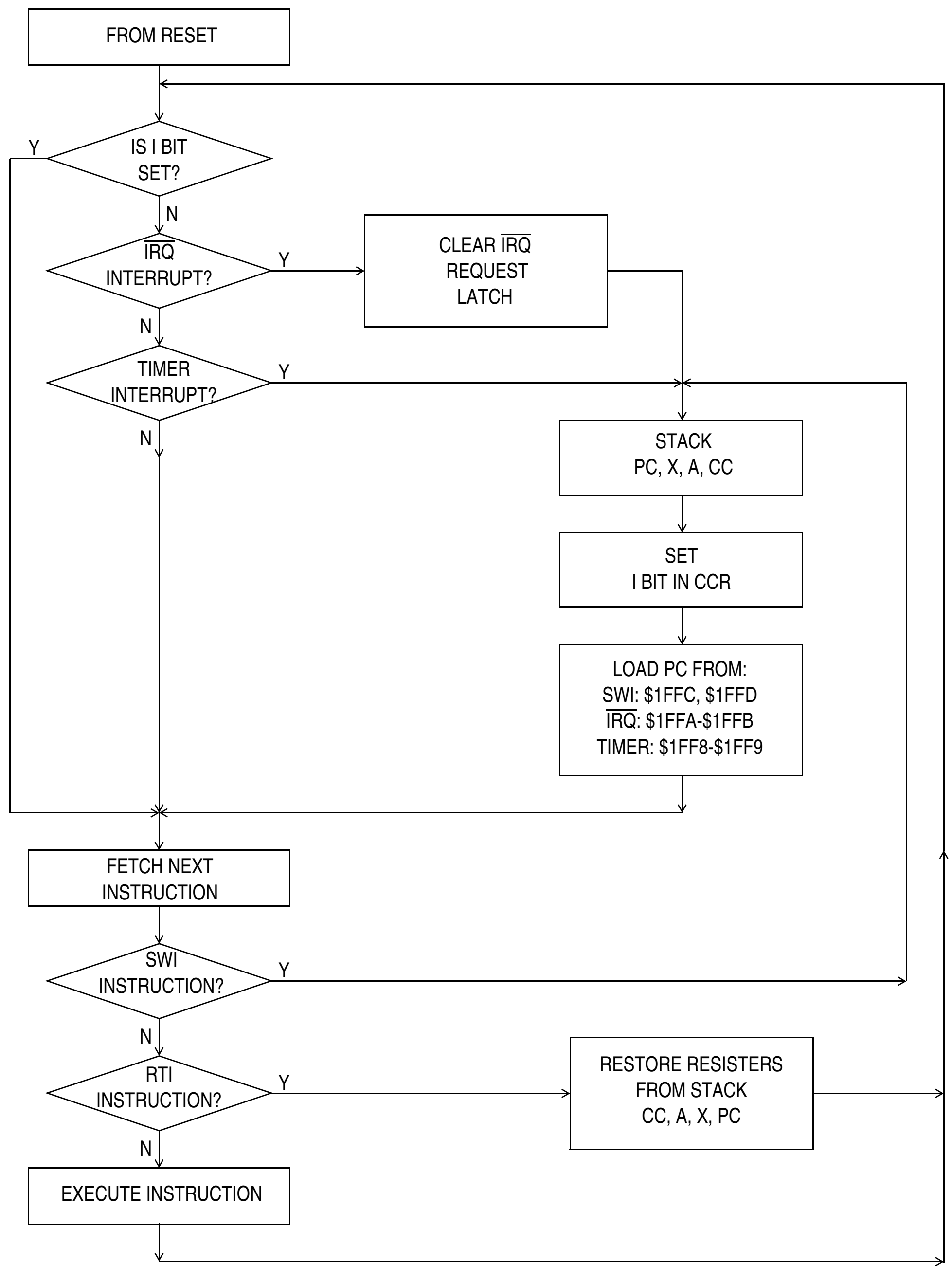


Figure 5-1. Interrupt Processing Flowchart

5.2 Interrupt Types

The interrupts fall into three categories: reset, software, and hardware.

5.2.1 Reset Interrupt Sequence

The reset function is not in the strictest sense an interrupt; however, it is acted upon in a similar manner as shown in Figure 5-1. A low-level input on the $\overline{\text{RESET}}$ pin or internally generated RST signal causes the program to vector to its starting address which is specified by the contents of memory locations $1FFE and $1FFF. The I bit in the condition code register is also set. The MCU is configured to a known state during this type of reset as previously described in Chapter 4 Resets.

5.2.2 Software Interrupt (SWI)

The SWI is an executable instruction. It is also a non-maskable interrupt since it is executed regardless of the state of the I bit in the CCR. As with any instruction, interrupts pending during the previous instruction will be serviced before the SWI opcode is fetched. The interrupt service routine address for the SWI instruction is specified by the contents of memory locations $1FFC and $1FFD.

5.2.3 Hardware Interrupts

All hardware interrupts are maskable by the I bit in the CCR. If the I bit is set, all hardware interrupts (internal and external) are disabled. Clearing the I bit enables the hardware interrupts. Four hardware interrupts are explained in the following subsections.

5.2.3.1 External Interrupt ($\overline{\text{IRQ}}$)

The $\overline{\text{IRQ}}/V_{PP}$ pin drives an asynchronous interrupt to the CPU. An edge detector flip-flop is latched on the falling edge of $\overline{\text{IRQ}}/V_{PP}$. If either the output from the internal edge detector flip-flop or the level on the $\overline{\text{IRQ}}/V_{PP}$ pin is low, a request is synchronized to the CPU to generate the IRQ interrupt. If the LEVEL bit in the mask option register is clear (edge-sensitive only), the output of the internal edge detector flip-flop is sampled and the input level on the $\overline{\text{IRQ}}/V_{PP}$ pin is ignored. The interrupt service routine address is specified by the contents of memory locations $1FFA and $1FFB. If the port A interrupts are enabled by the MOR, they generate external interrupts identically to the $\overline{\text{IRQ}}/V_{PP}$ pin.

NOTE

The internal interrupt latch is cleared nine internal clock cycles after the interrupt is recognized (immediately after location $1FFA is read). Therefore, another external interrupt pulse could be latched during the $\overline{\text{IRQ}}$ service routine.

Another interrupt will be serviced if the $\overline{\text{IRQ}}$ pin is still in a low state when the RTI in the service routine is executed.

5.2.3.2 Input Capture Interrupt

The input capture interrupt is generated by the 16-bit timer as described in Chapter 8 Capture/Compare Timer. The input capture interrupt flag is located in register TSR and its corresponding enable bit can be found in register TCR. The I bit in the CCR must be clear for the input capture interrupt to be enabled. The interrupt service routine address is specified by the contents of memory locations $1FF8 and $1FF9.

5.2.3.3 Output Compare Interrupt

The output compare interrupt is generated by a 16-bit timer as described in Chapter 8 Capture/Compare Timer. The output compare interrupt flag is located in register TSR and its corresponding enable bit can be found in register TCR. The I bit in the CCR must be clear for the output compare interrupt to be enabled. The interrupt service routine address is specified by the contents of memory locations $1FF8 and $1FF9.

5.2.3.4 Timer Overflow Interrupt

The timer overflow interrupt is generated by the 16-bit timer as described in Chapter 8 Capture/Compare Timer. The timer overflow interrupt flag is located in register TSR and its corresponding enable bit can be found in register TCR. The I bit in the CCR must be clear for the timer overflow interrupt to be enabled. This internal interrupt will vector to the interrupt service routine located at the address specified by the contents of memory locations $1FF8 and $1FF9.

Chapter 6
Input/Output Ports

6.1 Introduction

In the user mode, 20 bidirectional I/O lines are arranged as two 8-bit I/O ports (ports A and C), one 3-bit I/O port (port B), and one 1-bit I/O port (port D). These ports are programmable as either inputs or outputs under software control of the data direction registers (DDRs). Port D also contains one input-only pin.

6.2 Port A

Port A is an 8-bit bidirectional port, which does not share any of its pins with other subsystems (see Figure 6-1). The port A data register is located at address \$0000 and its data direction register (DDR) is located at address \$0004. The contents of the port A data register are indeterminate at initial power up and must be initialized by user software. Reset does not affect the data registers, but does clear the DDRs, thereby setting all of the port pins to input mode. Writing a 1 to a DDR bit sets the corresponding port pin to output mode. Port A has mask option register enabled interrupt capability with an internal pullup device

NOTE

The keyscan (pullup/interrupt) feature available on port A is NOT available in the ROM device, MC68HC05P6.

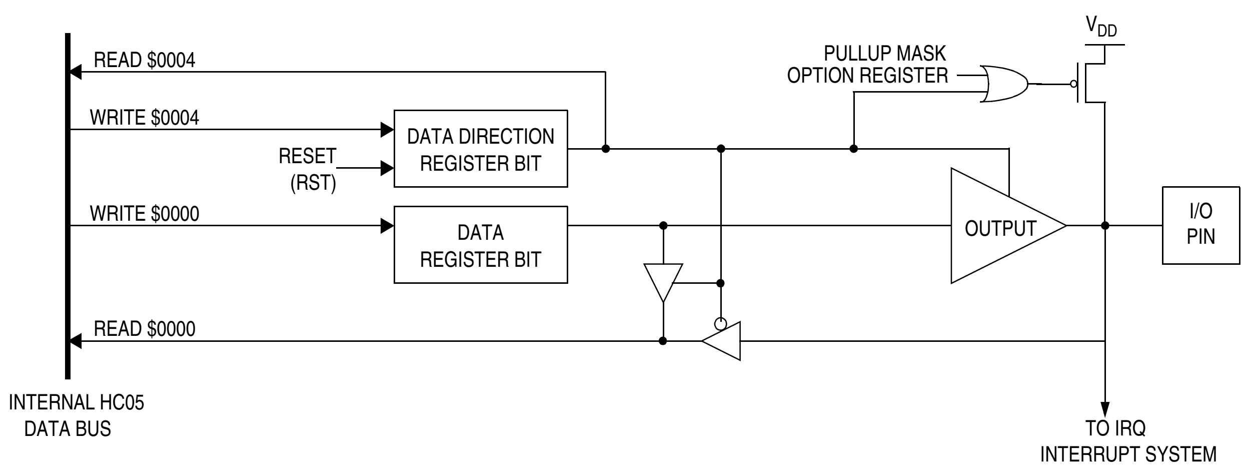


Figure 6-1. Port A I/O and Interrupt Circuitry

6.3 Port B

Port B is a 3-bit bidirectional port which can share pins PB5–PB7 with the SIOP communications subsystem. The port B data register is located at address $0001 and its data direction register (DDR) is located at address $0005. The contents of the port B data register are indeterminate at initial powerup and must be initialized by user software. Reset does not affect the data registers, but clears the DDRs, thereby setting all of the port pins to input mode. Writing a 1 to a DDR bit sets the corresponding port pin to output mode (see Figure 6-2).

Port B may be used for general I/O applications when the SIOP subsystem is disabled. The SPE bit in register SPCR is used to enable/disable the SIOP subsystem. When the SIOP subsystem is enabled, port B registers are still accessible to software. Writing to either of the port B registers while a data transfer is under way could corrupt the data. See Chapter 7 Serial Input/Output Port (SIOP) for a discussion of the SIOP subsystem.

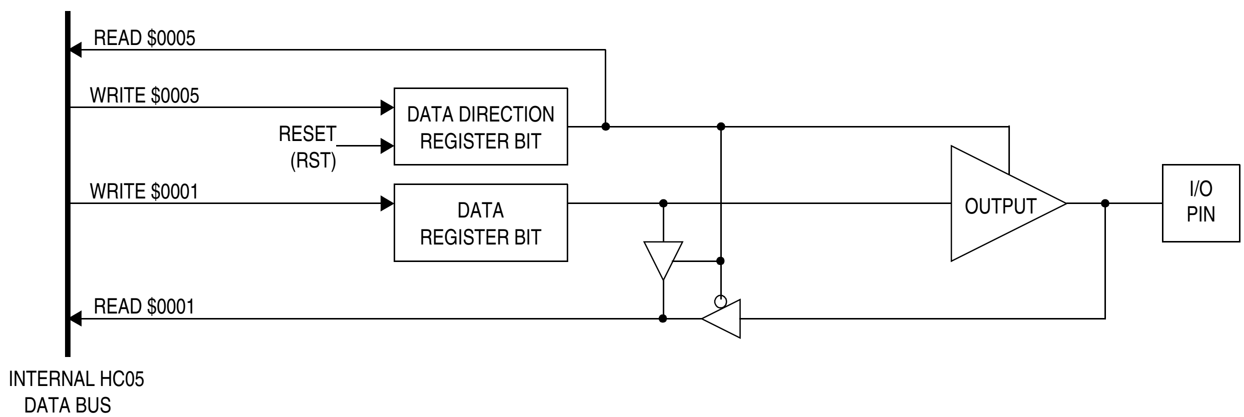


Figure 6-2. Port B I/O Circuitry

6.4 Port C

Port C is an 8-bit bidirectional port which can share pins PC3–PC7 with the A/D subsystem. The port C data register is located at address $0002 and its data direction register (DDR) is located at address $0006. The contents of the port C data register are indeterminate at initial powerup and must be initialized by user software. Reset does not affect the data registers, but clears the DDRs, thereby setting all of the port pins to input mode. Writing a 1 to a DDR bit sets the corresponding port pin to output mode (see Figure 6-3).

Port C may be used for general I/O applications when the A/D subsystem is disabled. The ADON bit in register ADSC is used to enable/disable the A/D subsystem. Care must be exercised when using pins PC0–PC2 while the A/D subsystem is enabled. Accidental changes to bits that affect pins PC3–PC7 in the data or DDR registers will produce unpredictable results in the A/D subsystem. See Chapter 9 Analog Subsystem.

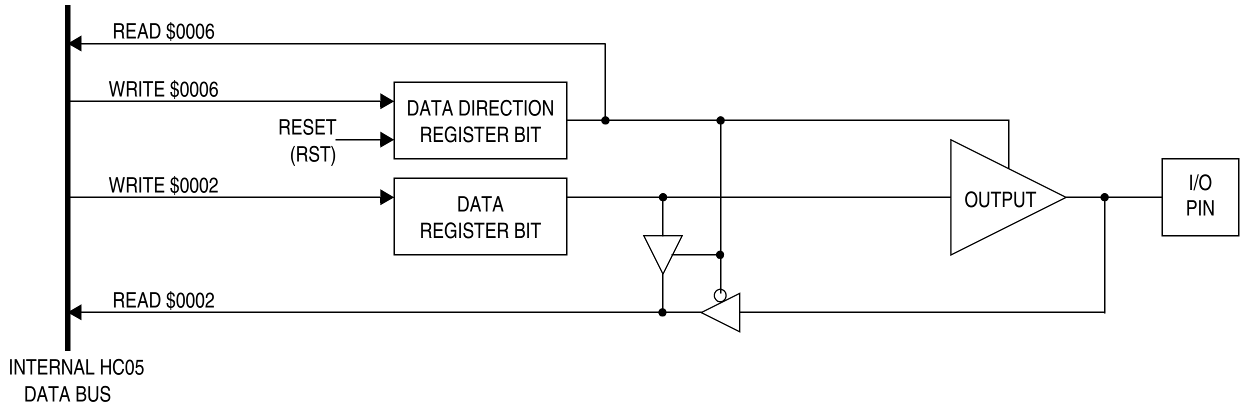


Figure 6-3. Port C I/O Circuitry

6.5 Port D

Port D is a 2-bit port with one bidirectional pin (PD5) and one input-only pin (PD7). Pin PD7 is shared with the 16-bit timer. The port D data register is located at address $0003 and its data direction register (DDR) is located at address $0007. The contents of the port D data register are indeterminate at initial powerup and must be initialized by user software. Reset does not affect the data registers, but clears the DDRs, thereby setting PD5 to input mode. Writing a 1 to DDR bit 5 sets PD5 to output mode (see Figure 6-4).

Port D may be used for general I/O applications regardless of the state of the 16-bit timer. Since PD7 is an input-only line, its state can be read from the port D data register at any time.

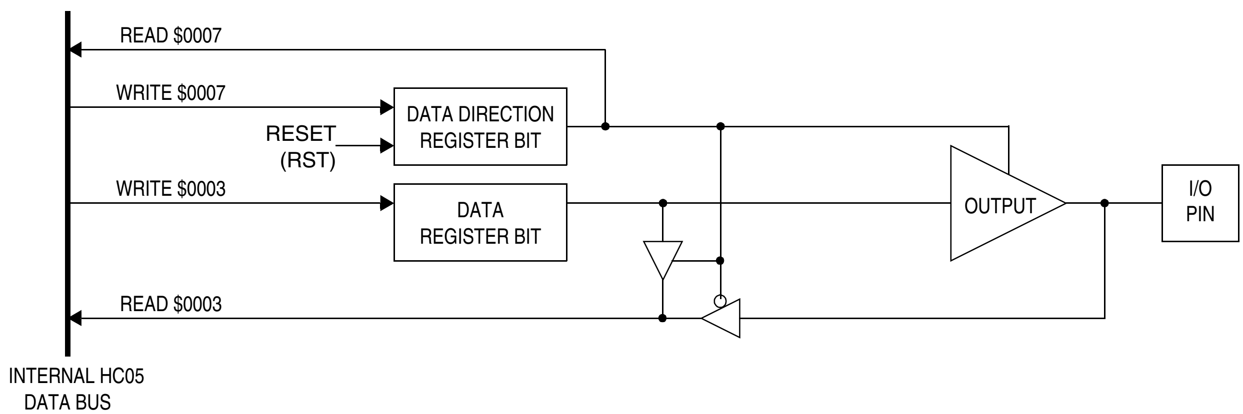


Figure 6-4. Port D I/O Circuitry

6.6 I/O Port Programming

Each pin on port A through port D (except pin 7 of port D) can be programmed as an input or an output under software control as shown in Table 6-1, Table 6-2, Table 6-3, and Table 6-4. The direction of a pin is determined by the state of its corresponding bit in the associated port data direction register (DDR). A pin is configured as an output if its corresponding DDR bit is set to a logic 1. A pin is configured as an input if its corresponding DDR bit is cleared to a logic 0.

Table 6-1. Port A I/O Functions

DDRA	I/O Pin Mode	Accesses to DDRA @ $0004	Accesses to Data Register @ $0000	
		Read/Write	Read	Write
0	IN, Hi-Z	DDRA0–DDRA7	I/O Pin	See Note
1	OUT	DDRA0–DDRA7	PA0–PA7	PA0–PA7

Note: Does not affect input, but stored to data register

Table 6-2. Port B I/O Functions

DDRB	I/O Pin Mode	Accesses to DDRB @ $0005	Accesses to Data Register @ $0001	
		Read/Write	Read	Write
0	IN, Hi-Z	DDRB5–DDRB7	I/O Pin	See Note
1	OUT	DDRB5–DDRB7	PB5–PB7	PB5–PB7

Note: Does not affect input, but stored to data register

Table 6-3. Port C I/O Functions

DDRC	I/O Pin Mode	Accesses to DDRC @ $0006	Accesses to Data Register @ $0002	
		Read/Write	Read	Write
0	IN, Hi-Z	DDRC0–DDRC7	I/O Pin	See Note
1	OUT	DDRC0–DDRC7	PC0–PC7	PC0–PC7

Note: Does not affect input, but stored to data register

Table 6-4. Port D I/O Functions

DDRD	I/O Pin Mode	Accesses to DDRD @ $0007	Accesses to Data Register @ $0003	
		Read/Write	Read	Write
0	IN, Hi-Z	DDRD5	I/O Pin	See Note 1
1	OUT	DDRD5	PD5	PD5

Notes:
1. Does not affect input, but stored to data register
2. PD7 is input only

NOTE

To avoid generating a glitch on an I/O port pin, data should be written to the I/O port data register before writing a logic 1 to the corresponding data direction register.

At power-on or reset, all DDRs are cleared, which configures all port pins as inputs. The DDRs are capable of being written to or read by the processor. During the programmed output state, a read of the data register will actually read the value of the output data latch and not the level on the I/O port pin.

Chapter 7
Serial Input/Output Port (SIOP)

7.1 Introduction

The simple synchronous serial I/O port (SIOP) subsystem is designed to provide efficient serial communications between peripheral devices or other MCUs. The SIOP is implemented as a 3-wire master/slave system with serial clock (SCK), serial data input (SDI), and serial data output (SDO). A block diagram of the SIOP is shown in Figure 7-1. A mask programmable option determines whether the SIOP is MSB or LSB first.

The SIOP subsystem shares its input/output pins with port B. When the SIOP is enabled (SPE bit set in register SCR), port B DDR and data registers are modified by the SIOP. Although port B DDR and data registers can be altered by application software, these actions could affect the transmitted or received data.

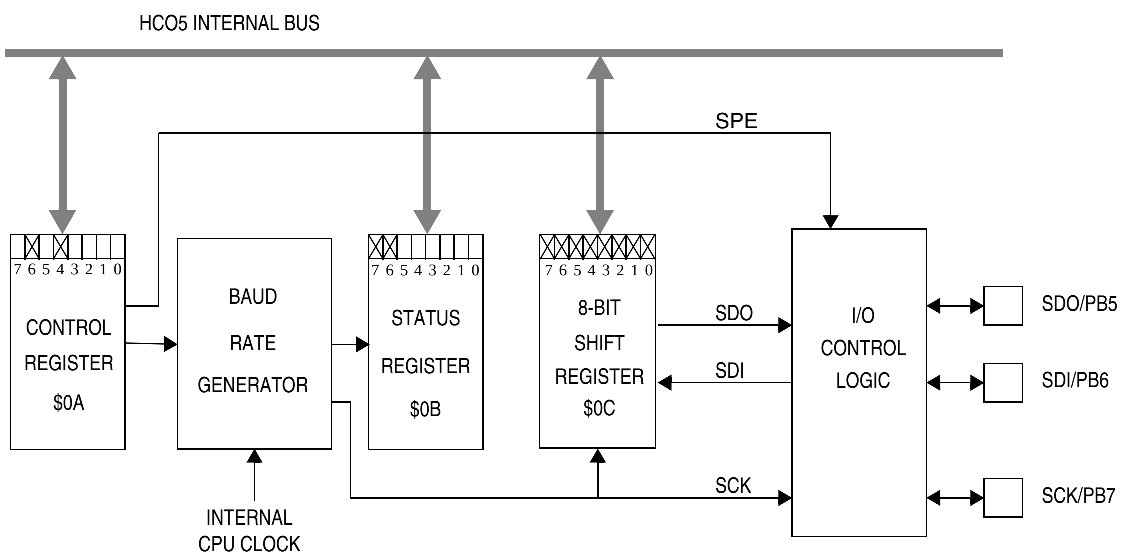


Figure 7-1. SIOP Block Diagram

7.2 SIOP Signal Format

The SIOP subsystem is software configurable for master or slave operation. No external mode selection inputs are available (for instance, slave select pin).

7.2.1 Serial Clock (SCK)

The state of the SCK output normally remains a logic 1 during idle periods between data transfers. The first falling edge of SCK signals the beginning of a data transfer. At this time, the first bit of received data may be presented at the SDI pin and the first bit of transmitted data is presented at the SDO pin (see Figure 7-2). Data is captured at the SDI pin on the rising edge of SCK. The transfer is terminated upon the eighth rising edge of SCK.

The master and slave modes of operation differ only by the sourcing of SCK. In master mode, SCK is driven from an internal source within the MCU. In slave mode, SCK is driven from a source external to the MCU. The SCK frequency is dependent upon the SPR0 and SPR1 bits located in the mask option register. Refer to 11.2 Mask Option Register for a description of available SCK frequencies.

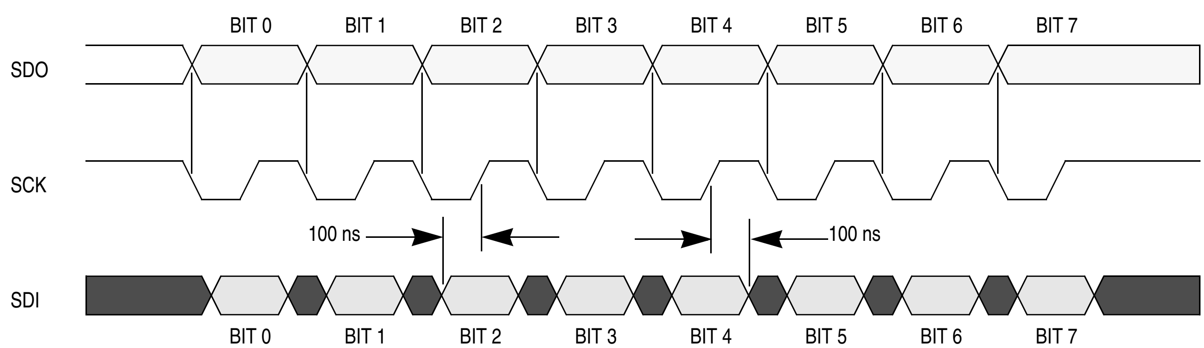


Figure 7-2. SIOP Timing Diagram

7.2.2 Serial Data Input (SDI)

The SDI pin becomes an input as soon as the SIOP subsystem is enabled. New data may be presented to the SDI pin on the falling edge of SCK.However, valid data must be present at least 100 nanoseconds before the rising edge of SCK and remain valid for 100 nanoseconds after the rising edge of SCK. See Figure 7-2.

7.2.3 Serial Data Output (SDO)

The SDO pin becomes an output as soon as the SIOP subsystem is enabled. Prior to enabling the SIOP, PB5 can be initialized to determine the beginning state. While the SIOP is enabled, PB5 cannot be used as a standard output since that pin is connected to the last stage of the SIOP serial shift register. Mask option register bit LSBF permits data to be transmitted in either the MSB first format or the LSB first format. Refer to 11.2 Mask Option Register for MOR LSBF programming information.

On the first falling edge of SCK, the first data bit will be shifted out to the SDO pin. The remaining data bits will be shifted out to the SDO pin on subsequent falling edges of SCK. The SDO pin will present valid data at least 100 nanoseconds before the rising edge of the SCK and remain valid for 100 nanoseconds after the rising edge of SCK. See Figure 7-2.

7.3 SIOP Registers

The SIOP is programmed and controlled by the SIOP control register (SCR) located at address $000A, the SIOP status register (SSR) located at address $000B, and the SIOP data register (SDR) located at address $000C.

7.3.1 SIOP Control Register (SCR)

This register is located at address $000A and contains two bits. Figure 7-3 shows the position of each bit in the register and indicates the value of each bit after reset.

Address: $000A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	SPE	0	MSTR	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

(shaded) = Unimplemented

Figure 7-3. SIOP Control Register (SCR)

SPE — Serial Peripheral Enable

When set, the SPE bit enables the SIOP subsystem such that SDO/PB5 is the serial data output, SDI/PB6 is the serial data input, and SCK/PB7 is a serial clock input in the slave mode or a serial clock output in the master mode. Port B DDR and data registers can be manipulated as usual (except for PB5); however, these actions could affect the transmitted or received data.

The SPE bit is readable at any time. However, writing to the SIOP control register while a transmission is in progress will cause the SPIF and DCOL bits in the SIOP status register (see below) to operate incorrectly. Therefore, the SIOP control register should be written once to enable the SIOP and then not written to until the SIOP is to be disabled. Clearing the SPE bit while a transmission is in progress will 1) abort the transmission, 2) reset the serial bit counter, and 3) convert the port B/SIOP port to a general-purpose I/O port. Reset clears the SPE bit.

MSTR — Master Mode Select

When set, the MSTR bit configures the serial I/O port for master mode. A transfer is initiated by writing to the SDR. Also, the SCK pin becomes an output providing a synchronous data clock dependent upon the oscillator frequency. When the device is in slave mode, the SDO and SDI pins do not change function. These pins behave exactly the same in both the master and slave modes.

The MSTR bit is readable and writeable at any time regardless of the state of the SPE bit. Clearing the MSTR bit will abort any transfers that may have been in progress. Reset clears the MSTR bit as well as the SPE bit, disabling the SIOP subsystem.

7.3.2 SIOP Status Register (SSR)

This register is located at address $000B and contains two bits. Figure 7-4 shows the position of each bit in the register and indicates the value of each bit after reset.

Address: $000B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SPIF	DCOL	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

(shaded) = Unimplemented

Figure 7-4. SIOP Status Register (SSR)

SPIF — Serial Port Interface Flag

SPIF is a read-only status bit that is set on the last rising edge of SCK and indicates that a data transfer has been completed. It has no effect on any future data transfers and can be ignored. The SPIF bit is cleared by reading the SSR followed by a read or write of the SDR. If the SPIF is cleared before the last rising edge of SCK, it will be set again on the last rising edge of SCK. Reset clears the SPIF bit.

DCOL — Data Collision

DCOL is a read-only status bit which indicates that an illegal access of the SDR has occurred. The DCOL bit will be set when reading or writing the SDR after the first falling edge of SCK and before SPIF is set. Reading or writing the SDR during this time will result in invalid data being transmitted or received.

The DCOL bit is cleared by reading the SSR (when the SPIF bit is set) followed by a read or write of the SDR. If the last part of the clearing sequence is done after another transfer has started, the DCOL bit will be set again. Reset clears the DCOL bit.

7.3.3 SIOP Data Register (SDR)

This register is located at address $000C and serves as both the transmit and receive data register. Writing to this register will initiate a message transmission if the SIOP is in master mode. The SIOP subsystem is not double buffered and any write to this register will destroy the previous contents. The SDR can be read at any time; however, if a transfer is in progress, the results may be ambiguous and the DCOL bit will be set. Writing to the SDR while a transfer is in progress can cause invalid data to be transmitted and/or received. Figure 7-5 shows the position of each bit in the register. This register is not affected by reset.

Address: $000C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
Reset:	Unaffected by reset							

Figure 7-5. Serial Port Data Register (SDR)

Chapter 8
Capture/Compare Timer

8.1 Introduction

This section describes the operation of the 16-bit capture/compare timer. Figure 8-1 shows the structure of the capture/compare subsystem.

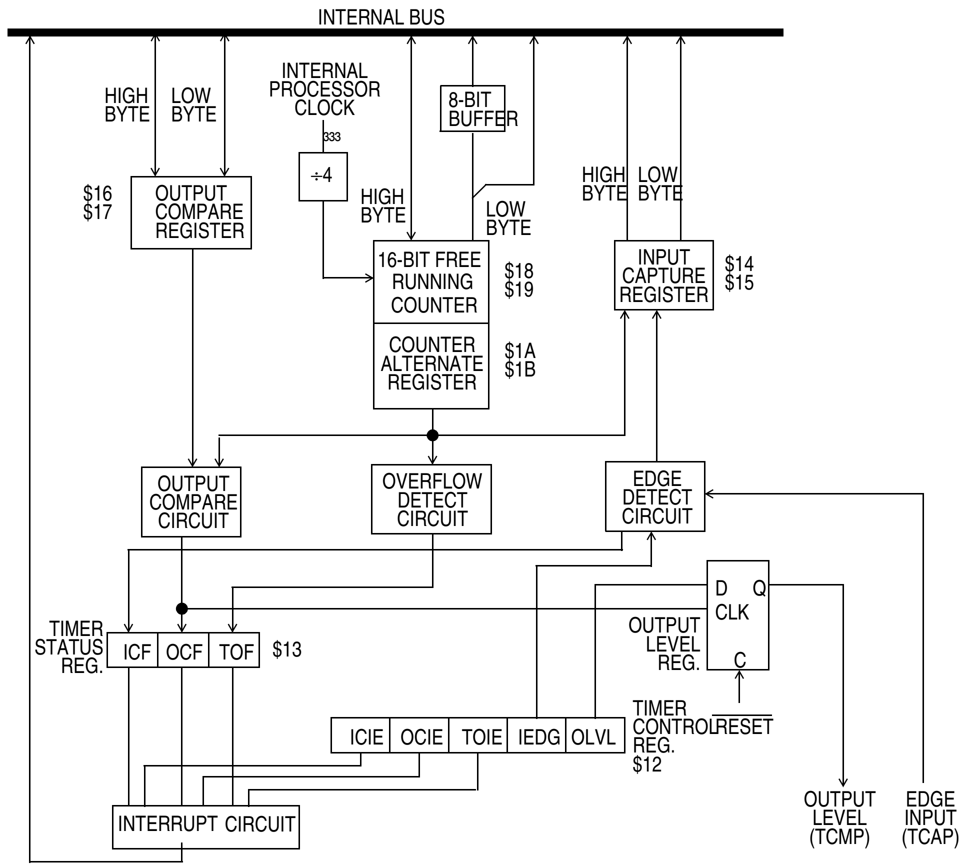


Figure 8-1. Capture/Compare Timer Block Diagram

8.2 Timer Operation

The core of the capture/compare timer is a 16-bit free-running counter. The counter provides the timing reference for the input capture and output compare functions. The input capture and output compare functions provide a means to latch the times at which external events occur, to measure input waveforms, and to generate output waveforms and timing delays. Software can read the value in the 16-bit free-running counter at any time without affecting the counter sequence.

Because of the 16-bit timer architecture, the I/O registers for the input capture and output compare functions are pairs of 8-bit registers.

Because the counter is 16 bits long and preceded by a fixed divide-by-4 prescaler, the counter rolls over every 262,144 internal clock cycles. Timer resolution with a 4-MHz crystal is 2 μs.

8.2.1 Input Capture

The input capture function is a means to record the time at which an external event occurs. When the input capture circuitry detects an active edge on the TCAP pin, it latches the contents of the timer registers into the input capture registers. The polarity of the active edge is programmable.

Latching values into the input capture registers at successive edges of the same polarity measures the period of the input signal on the TCAP pin. Latching values into the input capture registers at successive edges of opposite polarity measures the pulse width of the signal.

8.2.2 Output Compare

The output compare function is a means of generating an output signal when the 16-bit counter reaches a selected value. Software writes the selected value into the output compare registers. On every fourth internal clock cycle the output compare circuitry compares the value of the counter to the value written in the output compare registers. When a match occurs, the timer transfers the programmable output level bit (OLVL) from the timer control register to the TCMP pin.

The programmer can use the output compare register to measure time periods, to generate timing delays, or to generate a pulse of specific duration or a pulse train of specific frequency and duty cycle on the TCMP pin.

8.3 Timer I/O Registers

The following I/O registers control and monitor timer operation:

- Timer control register (TCR)
- Timer status register (TSR)
- Timer registers (TRH and TRL)
- Alternate timer registers (ATRH and ATRL)
- Input capture registers (ICRH and ICRL)
- Output compare registers (OCRH and OCRL)

8.3.1 Timer Control Register

The timer control register (TCR), shown in Figure 8-2, performs these functions:

- Enables input capture interrupts
- Enables output compare interrupts
- Enables timer overflow interrupts
- Controls the active edge polarity of the TCAP signal
- Controls the active level of the TCMP output

Address: $0012

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
Write:								
Reset:	0	0	0	0	0	0	U	0

= Unimplemented U = Undetermined

Figure 8-2. Timer Control Register (TCR)

ICIE — Input Capture Interrupt Enable

This read/write bit enables interrupts caused by an active signal on the TCAP pin. Resets clear the ICIE bit.

1 = Input capture interrupts enabled
0 = Input capture interrupts disabled

OCIE — Output Compare Interrupt Enable

This read/write bit enables interrupts caused by an active signal on the TCMP pin. Resets clear the OCIE bit.

1 = Output compare interrupts enabled
0 = Output compare interrupts disabled

TOIE — Timer Overflow Interrupt Enable

This read/write bit enables interrupts caused by a timer overflow. Reset clear the TOIE bit.

1 = Timer overflow interrupts enabled
0 = Timer overflow interrupts disabled

IEDG — Input Edge

The state of this read/write bit determines whether a positive or negative transition on the TCAP pin triggers a transfer of the contents of the timer register to the input capture register. Resets have no effect on the IEDG bit.

1 = Positive edge (low to high transition) triggers input capture
0 = Negative edge (high to low transition) triggers input capture

OLVL — Output Level

The state of this read/write bit determines whether a logic 1 or logic 0 appears on the TCMP pin when a successful output compare occurs. Resets clear the OLVL bit.

1 = TCMP goes high on output compare
0 = TCMP goes low on output compare

8.3.2 Timer Status Register

The timer status register (TSR), shown in Figure 8-3, contains flags to signal the following conditions:

- An active signal on the TCAP pin, transferring the contents of the timer registers to the input capture registers
- A match between the 16-bit counter and the output compare registers, transferring the OLVL bit to the TCMP pin
- A timer roll over from $FFFF to $0000

Address: $0013

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ICF	OCF	TOF	0	0	0	0	0
Write:								
Reset:	U	U	U	0	0	0	0	0

= Unimplemented U = Undetermined

Figure 8-3. Timer Status Register (TSR)

ICF — Input Capture Flag

The ICF bit is set automatically when an edge of the selected polarity occurs on the TCAP pin. Clear the ICF bit by reading the timer status register with ICF set and then reading the low byte ($0015) of the input capture registers. Resets have no effect on ICF.

OCF — Output Compare Flag

The OCF bit is set automatically when the value of the timer registers matches the contents of the output compare registers. Clear the OCF bit by reading the timer status register with OCF set and then reading the low byte ($0017) of the output compare registers. Resets have no effect on OCF.

TOF — Timer Overflow Flag

The TOF bit is set automatically when the 16-bit counter rolls over from $FFFF to $0000. Clear the TOF bit by reading the timer status register with TOF set, and then reading the low byte ($0019) of the timer registers. Resets have no effect on TOF.

8.3.3 Timer Registers

The timer registers (TRH and TRL), shown in Figure 8-4, contains the current high and low bytes of the 16-bit counter. Reading TRH before reading TRL causes TRL to be latched until TRL is read. Reading TRL after reading the timer status register clears the timer overflow flag (TOF). Writing to the timer registers has no effect.

Address: TRH — $0018

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TRH7	TRH6	TRH5	TRH4	TRH3	TRH2	TRH1	TRH0
Write								
Reset:	1	1	1	1	1	1	1	1

Address: TRL — $0019

	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	1	1	1	1	1	1	0	0

= Unimplemented

Figure 8-4. Timer Registers (TRH and TRL)

8.3.4 Alternate Timer Registers

The alternate timer registers (ATRH and ATRL), shown in Figure 8-5, contain the current high and low bytes of the 16-bit counter. Reading ATRH before reading ATRL causes ATRL to be latched until ATRL is read. Reading ATRL has no effect on the timer overflow flag (TOF). Writing to the alternate timer registers has no effect.

Address: ATRH — $001A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ACRH7	ACRH6	ACRH5	ACRH4	ACRH3	ACRH2	ACRH1	ACRH0
Write:								
Reset:	1	1	1	1	1	1	1	1

Address: ATRL — $001B

	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	1	1	1	1	1	1	0	0

= Unimplemented

Figure 8-5. Alternate Timer Registers (ATRH and ATRL)

NOTE

To prevent interrupts from occurring between readings of ATRH and ATRL, set the interrupt flag in the condition code register before reading ATRH, and clear the flag after reading ATRL.

8.3.5 Input Capture Registers

When a selected edge occurs on the TCAP pin, the current high and low bytes of the 16-bit counter are latched into the input capture registers. Reading ICRH before reading ICRL inhibits further capture until ICRL is read. Reading ICRL after reading the status register clears the input capture flag (ICF). Writing to the input capture registers has no effect.

Address: ICRH — $0014

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ICRH7	ICRH6	ICRH5	ICRH4	ICRH3	ICRH2	ICRH1	ICRH0
Write:								

Unaffected by reset

Address: ICRL — $0015

	Bit 7	6	5	4	3	2	1	Bit 0
Write:								

Unaffected by reset

[ ] = Unimplemented

Figure 8-6. Input Capture Registers (ICRH and ICRL)

NOTE

To prevent interrupts from occurring between readings of ICRH and ICRL, set the interrupt flag in the condition code register before reading ICRH, and clear the flag after reading ICRL.

8.3.6 Output Compare Registers

When the value of the 16-bit counter matches the value in the output compare registers, the planned TCMP pin action takes place. Writing to OCRH before writing to OCRL inhibits timer compares until OCRL is written. Reading or writing to OCRL after the timer status register clears the output compare flag (OCF).

Address: OCRH — $0016

	Bit 7	6	5	4	3	2	1	Bit 0
Write: Read:	OCRH7	OCRH6	OCRH5	OCRH4	OCRH3	OCRH2	OCRH1	OCRH0

Unaffected by reset

Address: OCRL — $0017

	Bit 7	6	5	4	3	2	1	Bit 0
Read:								

Unaffected by reset

Figure 8-7. Output Compare Registers (OCRH and OCRL)

To prevent OCF from being set between the time it is read and the time the output compare registers are updated, use this procedure:

1. Disable interrupts by setting the I bit in the condition code register.
2. Write to OCRH. Compares are now inhibited until OCRL is written.
3. Clear bit OCF by reading timer status register (TSR).
4. Enable the output compare function by writing to OCRL.
5. Enable interrupts by clearing the I bit in the condition code register.

8.4 Timer During Wait/Halt Mode

The CPU clock halts during the wait (or halt) mode, but the timer remains active. If interrupts are enabled, a timer interrupt will cause the processor to exit the wait mode.

8.5 Timer During Stop Mode

In the stop mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If STOP is exited by RESET, the counters are forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pins, the input capture detect circuit is armed. This does not set any timer flags or wake up the MCU, but if an interrupt is used to exit stop mode, there is an active input capture flag and data from the first valid edge that occurred during the stop mode. If reset is used to exit stop mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

Chapter 9
Analog Subsystem

9.1 Introduction

The MC68HC705P6A includes a 4-channel, multiplexed input, 8-bit, successive approximation analog-to-digital (A/D) converter. The A/D subsystem shares its inputs with port C pins PC3–PC7.

9.2 Analog Section

The following paragraphs describe the operation and performance of analog modules within the analog subsystem.

9.2.1 Ratiometric Conversion

The A/D converter is ratiometric, with pin V_{REFH} supplying the high reference voltage. Applying an input voltage equal to V_{REFH} produces a conversion result of $FF (full scale). Applying an input voltage equal to V_{SS} produces a conversion result of $00. An input voltage greater than V_{REFH} will convert to $FF with no overflow indication. For ratiometric conversions, V_{REFH} should be at the same potential as the supply voltage being used by the analog signal being measured and referenced to V_{SS}.

9.2.2 Reference Voltage (V_{REFH})

The reference supply for the A/D converter shares pin PC7 with port C. The low reference is tied to the V_{SS} pin internally. V_{REFH} can be any voltage between V_{SS} and V_{DD}; however, the accuracy of conversions is tested and guaranteed only for $V_{REFH} = V_{DD}$.

9.2.3 Accuracy and Precision

The 8-bit conversion result is accurate to within ±1 1/2 LSB, including quantization; however, the accuracy of conversions is tested and guaranteed only with external oscillator operation.

9.3 Conversion Process

The A/D reference inputs are applied to a precision digital-to-analog converter. Control logic drives the D/A and the analog output is successively compared to the selected analog input which was sampled at the beginning of the conversion cycle. The conversion process is monotonic and has no missing codes.

9.4 Digital Section

The following paragraphs describe the operation and performance of digital modules within the analog subsystem.

9.4.1 Conversion Times

Each input conversion requires 32 internal clock cycles, which must be at a frequency equal to or greater than 1 MHz.

9.4.2 Internal versus External Oscillator

If the internal clock is 1 MHz or greater (i.e., external oscillator 2 MHz or greater), the internal RC oscillator must be turned off and the external oscillator used as the conversion clock.

If the MCU internal clock frequency is less than 1 MHz (2 MHz external oscillator), the internal RC oscillator (approximately 1.5 MHz) must be used for the A/D converter clock. The internal RC clock is selected by setting the ADRC bit in the ADSC register.

When the internal RC oscillator is being used, these limitations apply:

1. Since the internal RC oscillator is running asynchronously with respect to the internal clock, the conversion complete bit (CC) in register ADSC must be used to determine when a conversion sequence has been completed.
2. Electrical noise will slightly degrade the accuracy of the A/D converter. The A/D converter is synchronized to read voltages during the quiet period of the clock driving it. Since the internal and external clocks are not synchronized, the A/D converter will occasionally measure an input when the external clock is making a transition.

9.4.3 Multi-Channel Operation

An input multiplexer allows the A/D converter to select from one of four external analog signals. Port C pins PC3 through PC6 are shared with the inputs to the multiplexer.

9.5 A/D Status and Control Register (ADSC)

The ADSC register reports the completion of A/D conversion and provides control over oscillator selection, analog subsystem power, and input channel selection. See Figure 9-1.

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CC	ADRC	ADON	0	0	CH2	CH1	CH0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented (shaded: bit 7, 4 and 3 write)

Figure 9-1. A/D Status and Control Register (ADSC)

CC — Conversion Complete

This read-only status bit is set when a conversion sequence has completed and data is ready to be read from the ADC register. CC is cleared when the ADSC is written to or when data is read from the ADC register. Once a conversion has been started, conversions of the selected channel will continue every 32 internal clock cycles until the ADSC register is written to again. During continuous conversion operation, the ADC register will be updated with new data, and the CC bit set every 32 internal clock cycles. Also, data from the previous conversion will be overwritten regardless of the state of the CC bit.

ADRC — RC Oscillator Control

When ADRC is set, the A/D subsystem operates from the internal RC oscillator instead of the internal clock. The RC oscillator requires a time, t_{RCON}, to stabilize before accurate conversion results can be obtained. See 9.2.2 Reference Voltage (VREFH) for more information.

ADON — A/D Subsystem On

When the A/D subsystem is turned on (ADON = 1), it requires a time, t_{ADON}, to stabilize before accurate conversion results can be attained.

CH2–CH0 — Channel Select Bits

CH2, CH1, and CH0 form a 3-bit field which is used to select an input to the A/D converter. Channels 0–3 correspond to port C input pins PC6–PC3. Channels 4–6 are used for reference measurements. Channel 7 is reserved. If a conversion is attempted with channel 7 selected, the result will be $00. Table 9-1 lists the inputs selected by bits CH0-CH3.

If the ADON bit is set and an input from channels 0–4 is selected, the corresponding port C pin's DDR bit will be cleared (making that port C pin an input). If the port C data register is read while the A/D is on and one of the shared input channels is selected using bit CH0–CH2, the corresponding port C pin will read as a logic 0. The remaining port C pins will read normally. To digitally read a port C pin, the A/D subsystem must be disabled (ADON = 0), or input channels 5–7 must be selected.

Table 9-1. A/D Multiplexer Input Channel Assignments

Channel	Signal
0	AD0 — port C, bit 6
1	AD1 — port C, bit 5
2	AD2 — port C, bit 4
3	AD3 — port C, bit 3
4	V_{REFH} — port C, bit 7
5	$(V_{REFH} + V_{SS})/2$
6	V_{SS}
7	Reserved for factory test

9.6 A/D Conversion Data Register (ADC)

This register contains the output of the A/D converter. See Figure 9-2.

Address: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Write:								
Reset:	Unaffected by reset							

[shaded box] = Unimplemented

Figure 9-2. A/D Conversion Value Data Register (ADC)

9.7 A/D Subsystem Operation during Halt/Wait Modes

The A/D subsystem continues normal operation during wait and halt modes. To decrease power consumption during wait or halt mode, the ADON and ADRC bits in the A/D status and control register should be cleared if the A/D subsystem is not being used.

9.8 A/D Subsystem Operation during Stop Mode

When stop mode is enabled, execution of the STOP instruction will terminate all A/D subsystem functions. Any pending conversion is aborted. When the oscillator resumes operation upon leaving stop mode, a finite amount of time passes before the A/D subsystem stabilizes sufficiently to provide conversions at its rated accuracy. The delays built into the MC68HC705P6A when coming out of stop mode are sufficient for this purpose. No explicit delays need to be added to the application software.

Chapter 10
EPROM

10.1 Introduction

The user EPROM consists of 48 bytes of user page zero EPROM from $0020 to $004F, 4608 bytes of user EPROM from $0100 to $12FF, the two MOR reset values located at $1EFF and $1F00, and 16 bytes of user vectors EPROM from $1FF0 to $1FFF. The bootloader ROM and vectors are located from $1F01 to $1FEF.

10.2 EPROM Erasing

NOTE

Only parts packaged in a windowed package may be erased. Others are one-time programmable and may not be erased by UV exposure.

The MC68HC705P6A can be erased by exposure to a high-intensity ultraviolet (UV) light with a wavelength of 2537 angstroms. The recommended dose (UV intensity multiplied by exposure time) is 15 Ws/cm^2. UV lamps without shortwave filters should be used, and the EPROM device should be positioned about one inch from the UV lamp. An erased EPROM byte will read as $00.

10.3 EPROM Programming Sequence

The bootloader software goes through a complete write cycle of the EPROM including the MOR. This is followed by a verify cycle which continually branches in a loop if an error is found. A sample routine to program a byte of EPROM is shown in Table 10-1.

NOTE

To avoid damage to the MCU, V_{DD} *must be applied to the MCU before* V_{PP}*.*

10.4 EPROM Registers

Three registers are associated with the EPROM: the EPROM programming register (EPROG) and the two mask option registers (MOR). The EPROG register controls the actual programming of the EPROM bytes and the MOR. The MOR registers control the six mask options found on the ROM version of this MCU (MC68HC05P6), the EPROM security feature, and eight additional port A interrupt options.

10.5 EPROM Programming Register (EPROG)

This register is used to program the EPROM array. Only the ELAT and EPGM bits are available. Table 10-1 shows the location of each bit in the EPROG register and the state of these bits coming out of reset. All the bits in the EPROG register are cleared by reset.

Address $001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	ELAT	0	EPGM
Write:								
Reset:	0	0	0	0	0	0	0	0

(Shaded write cells for bits 7–3 and bit 1) = Unimplemented

Figure 10-1. EPROM Programming Register (EPROG)

EPGM — EPROM Program Control

If the EPGM bit is set, programming power is applied to the EPROM array. If the EPGM bit is cleared, programming power is removed from the EPROM array. The EPGM bit cannot be set unless the ELAT bit is set already.

Whenever the ELAT bit is cleared, the EPGM bit is cleared also. Both the EPGM and the ELAT bit cannot be set using the same write instruction. Any attempt to set both the EPGM and ELAT bit on the same write instruction cycle will result in the ELAT bit being set and the EPGM bit being cleared. The EPGM bit is a read-write bit and can be read at any time. The EPGM bit is cleared by reset.

ELAT— EPROM Latch Control

If the ELAT bit is set, the EPROM address and data bus are configured for programming to the array. If the ELAT bit is cleared, the EPROM address and data bus are configured for normal reading of data from the array. When the ELAT bit is set, the address and data bus are latched in the EPROM array when a subsequent write to the array is made. Data in the EPROM array cannot be read if the ELAT bit is set.

Whenever the ELAT bit is cleared, the EPGM bit is cleared also. Both the EPGM and the ELAT bit cannot be set using the same write instruction. Any attempt to set both the EPGM and ELAT bit on the same write instruction cycle will result in the ELAT bit being set and the EPGM bit being cleared. The ELAT bit is a read-write bit and can be read at any time. The ELAT bit is cleared by reset.

To program a byte of EPROM, manipulate the EPROG register as follows:

1. Set the ELAT bit in the EPROG register.
2. Write the desired data to the desired EPROM address.
3. Set the EPGM bit in the EPROG register for the specified programming time, t_{EPGM}.
4. Clear the ELAT and EPGM bits in the EPROG register.

This sequence is also shown in the sample program listing in Table 10-1.

Table 10-1. EPROM Programming Routine

001C		EPROG	EQU $1C	PROGRAMMING REG
0055		DATA	EQU $55	DATA VALUE
0700		EPROM	EQU $700	A SAMPLE EPROM ADX
0000		EPGM	EQU $00	EPGM BIT IN EPROG REG
00D0		ORG	$D0	
00D0	A6 02		LDA #$04	SET LAT BIT IN EPROG
00D2	B7 1C		STA EPROG	
00D4	A6 55		LDA #DATA	DATA BYTE
00D6	C7 07 00		STA EPROM	WRITE IT TO EPROM LOC
00D9	10 1C		BSET EPGM, EPROG	TURN ON PGM VOLTAGE
00DB	AD 03		BSR DELAY	WAIT 4 ms MINIMUM
00DD	3F 1C		CLR EPROG	CLR LAT AND PGM BITS
00DF	81		RTS	

10.6 EPROM Bootloader

Three port pins are associated with bootloader control functions: PC3, PC4, and PC6. Table 10-2 summarizes their functionality.

Table 10-2. Bootloader Control Pins

PC6	PC4	PC3	Mode
1	1	1	Program/verify
1	1	0	Verify only
1	0	0	Dump MCU EPROM to port A

10.7 Programming from an External Memory Device

In this programming mode, PC5 must be connected to V_{SS}. PC4 and PC3 are used to select the programming mode. The programming circuit shown in Figure 10-2 uses an external 12-bit counter to address the memory device containing the code to be copied. This counter requires a clock and a reset function. The 12-bit counter can address up to 4 Kbytes of memory, which means that a port pin has to be used to address the remaining 4 K of the 8-K memory space.

The following procedure explains how to use the programming circuit shown in Figure 10-2 to copy a user program from an external memory device into the MCU's EPROM:

1. Program a 2764-type EPROM device with the desired instructions and data. Code programmed into the 2764 must appear at the same addresses desired in the MC68HC705P6A. Therefore, the page zero code must start at $0020 and end at $004F, the main body of code must start at $0100 and end at $12FF, and the user vectors must start at $1FF0 and end at $1FFF.

NOTE

The MOR data must appear at $1EFF and $1F00.

2. Install the programmed 2764 device into the programming circuit.
3. Install the MC68HC705P6A to be programmed into the programming circuit.
4. Set the PROGRAM and/or VERIFY switches for the desired operation (an open switch is the active state) and close the RESET switch to hold the MCU in reset.
5. Make sure that the V_{PP} source is OFF.
6. Apply the V_{DD} source to the programming circuit.
7. Apply the V_{PP} source to the programming circuit.
8. Open the RESET switch to allow the MCU to come out of reset and begin execution of the software in its internal bootloader ROM.
9. Wait for programming and/or verification to complete (about 40 seconds). The PROGRAM LED will light during programming and the VERIFY LED will light if verification was requested and was successful.
10. When complete, close the RESET switch to force the MCU into the reset state.
11. Turn off the V_{PP} source.
12. Turn off the V_{DD} source.
13. Remove device(s).

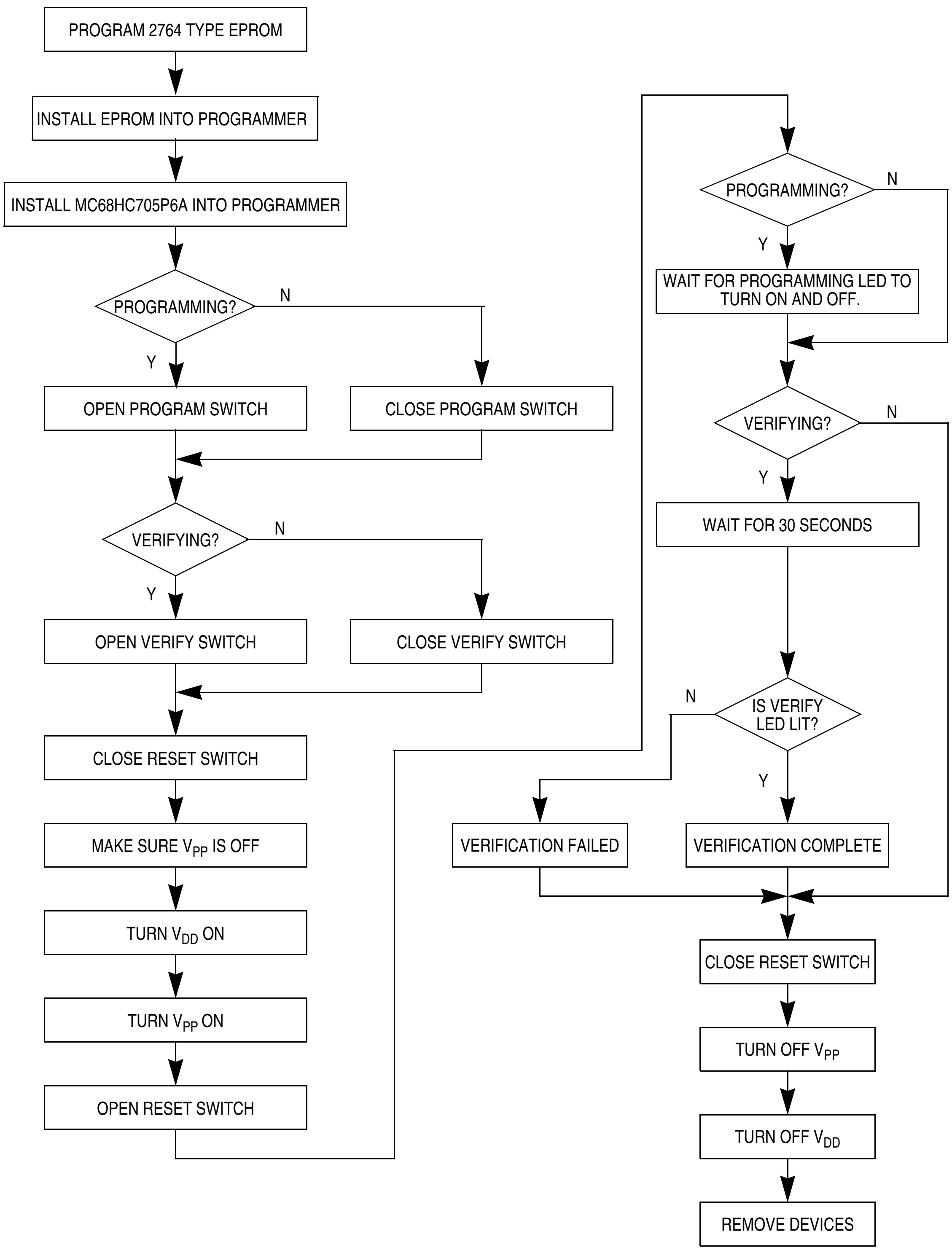


Figure 10-2. MC68HC705P6A EPROM Programming Flowchart

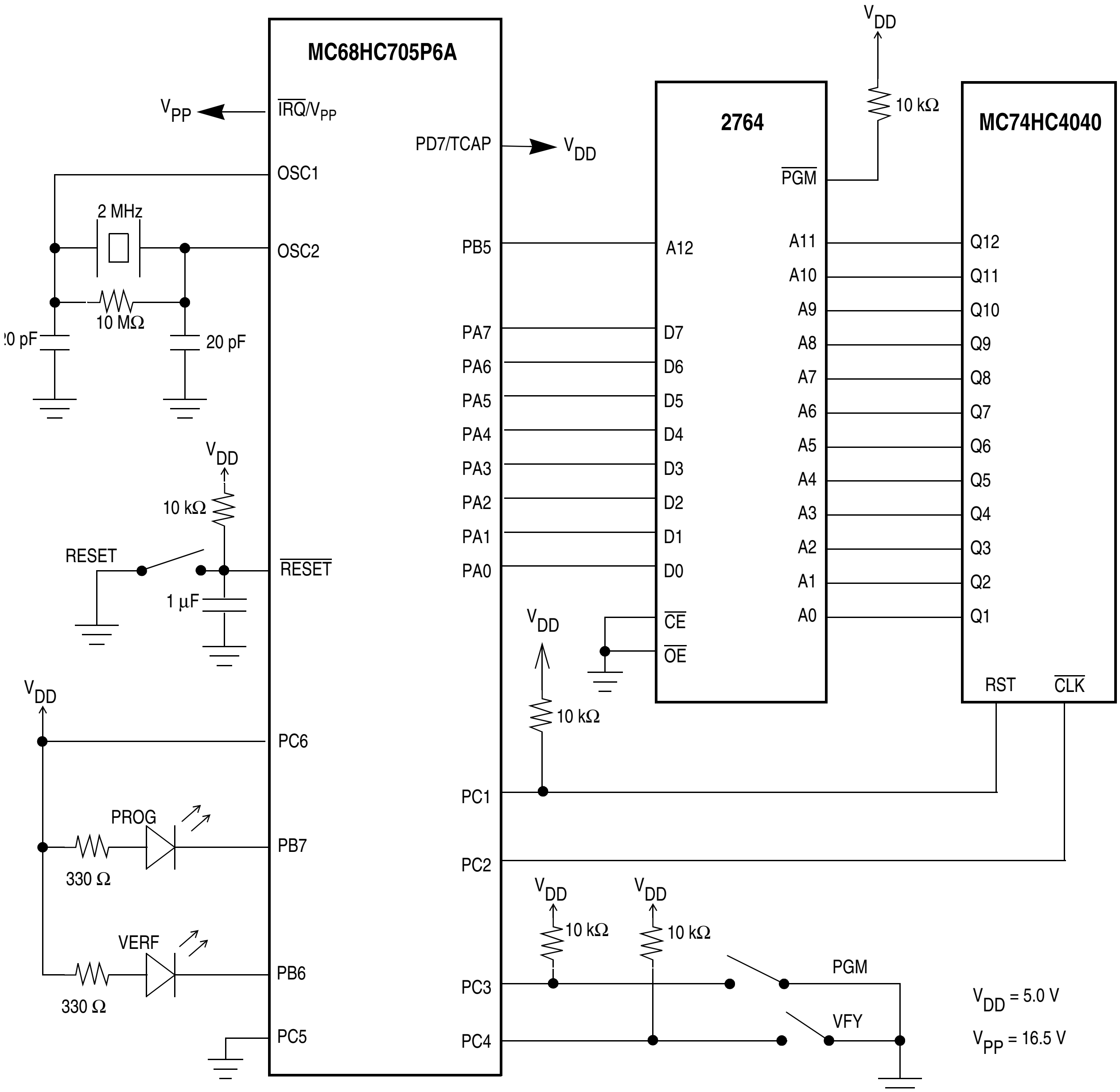


Figure 10-3. MC68HC705P6A EPROM Programming Schematic Diagram

Chapter 11
Mask Option Register (MOR)

11.1 Introduction

The mask option register (MOR) contains two bytes of EPROM used to enable or disable each of the features controlled by mask options on the MC68HC05P6 (a ROM version of the MC68HC705P6A).

The seven programmable options on the MC68HC705P6A are:

1. COP watchdog timer (enable or disable)
2. $\overline{\text{IRQ}}$ triggering (edge- or edge- and level-sensitive)
3. SIOP data bit order (most significant bit or least significant bit first)
4. SIOP clock rate (OSC divided by 8, 16, 32, or 64)
5. Stop instruction mode (stop mode or halt mode)
6. Secure EPROM from external reading
7. Keyscan interrupt/pullups on PA0–PA7

11.2 Mask Option Register

Mask options are programmed into the mask option register (MOR) by the firmware in the bootloader ROM. See Figure 11-1.

Address: $1EFF

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PA7PU	PA6PU	PA5PU	PA4PU	PA3PU	PA2PU	PA1PU	PA0PU
Erased State:	0	0	0	0	0	0	0	0

Address: $1F00

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SECURE		SWAIT	SPR1	SPR0	LSBF	LEVEL	COP
Erased State:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-1. Mask Option Register (MOR)

COP — COP Watchdog Enable

Setting the COP bit will enable the COP watchdog timer. The COP will reset the MCU if the timeout period is reached before the COP watchdog timer is cleared by the application software *and* the voltage applied to the $\overline{\text{IRQ}}/V_{PP}$ pin is between V_{SS} and V_{DD}. Clearing the COP bit will disable the COP watchdog timer regardless of the voltage applied to the $\overline{\text{IRQ}}/V_{PP}$ pin.

LEVEL — $\overline{IRQ}$ Edge Sensitivity

If the LEVEL bit is clear, the $\overline{IRQ}/V_{PP}$ pin will only be sensitive to the falling edge of the signal applied to the $\overline{IRQ}/V_{PP}$ pin. If the LEVEL bit is set, the $\overline{IRQ}/V_{PP}$ pin will be sensitive to both the falling edge of the input signal and the logic low level of the input signal on the $\overline{IRQ}/V_{PP}$ pin.

LSBF — SIOP Least Significant Bit First

If the LSBF bit is set, the serial data to and from the SIOP will be transferred least significant bit first. If the LSBF bit is clear, the serial data to and from the SIOP will be transferred most significant bit first.

SPR0 and SPR1 — SIOP Clock Rate

The SPR0 and SPR1 bits determine the clock rate used to transfer the serial data to and from the SIOP. The various clock rates available are given in Table 11-1.

Table 11-1. SIOP Clock Rate

SPR1	SPR0	SIOP Master Clock
0	0	$f_{osc} \div 64$
0	1	$f_{osc} \div 32$
1	0	$f_{osc} \div 16$
1	1	$f_{osc} \div 8$

SWAIT — STOP Instruction Mode

Setting the SWAIT bit will prevent the STOP instruction from stopping the on-board oscillator. Clearing the SWAIT bit will permit the STOP instruction to stop the on-board oscillator and place the MCU in stop mode. Executing the STOP instruction when SWAIT is set will place the MCU in halt mode. See 3.4.1 STOP Instruction for additional information.

SECURE — Security State[(1)]

If SECURE bit is set, the EPROM is locked.

PA(0:7)PU — Port A Pullups/Interrupt Enable/Disable

If any PA(0:7)PU is selected, that pullup/interrupt is enabled. The interrupt sensitivity will be selected via the LEVEL bit in the same way as the $\overline{IRQ}$ pin.

NOTE

*The port A pullup/interrupt function is **NOT** available on the ROM device, MC68HC05P6.*

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the EPROM/OTPROM difficult for unauthorized users.

11.3 MOR Programming

The contents of the MOR should be programmed in bootloader mode using the hardware shown in Figure 10-2. MC68HC705P6A EPROM Programming Flowchart. In order to allow programming, all the implemented bits in the MOR are essentially read-write bits in bootloader mode as shown in Figure 11-1.

The programming of the MOR is the same as user EPROM.

1. Set the ELAT bit in the EPROG register.
2. Write the desired data to the desired MOR address.
3. Set the EPGM bit in the EPROG.
4. Wait for the programming time (t_{EPGM}).
5. Clear the ELAT and EPGM bits in the EPROG.
6. Remove the programming voltage from the $\overline{IRQ}/V_{PP}$ pin.

A sample routine to program a byte of EPROM is shown in Table 11-2.

Once the MOR bits have been programmed, the options are not loaded into the MOR registers until the part is reset.

Table 11-2. MOR Programming Routine

```
001C                EPROG   EQU  $1C           PROGRAMMING REG
00FF                DATA2   EQU  $FF           SAMPLE MOR VALUES
0023                DATA1   EQU  #23
1EFF                MOR2    EQU  $1EFF         MOPR ADDRESSES
1F00                MOR1    EQU  $1F00
0000                EPGM    EQU  $00           EPGM BIT IN EPROG REG

00E0                        ORG  $E0

00E0   A6 04                LDA  #$04          SET ELAT BIT
00E2   B7 1C                STA  EPROG         IN EPGM REG AT $1C
00E4   A6 FF                LDA  #DATA2        DATA BYTE
00E6   C7 1E FF             STA  MOR2          WRITE IT TO MOR LOC
00E9   12 1C                BSET EPGM,EPROG    TURN ON PGM VOLTAGE
00EB   AD 03                BSR  DELAY         WAIT 4 ms MINIMUM
00ED   3F 1C                CLR  EPROG         CLR EPGM REGISTER
00EF   81                   RTS
```

Chapter 12
Central Processor Unit (CPU) Core

12.1 Introduction

The MC68HC705P6A has an 8-K memory map. Therefore, it uses only the lower 13 bits of the address bus. In the following discussion, the upper three bits of the address bus can be ignored. Also, the STOP instruction can be modified to place the MCU in either the normal stop mode or the halt mode by means of a MOR bit. All other instructions and registers behave as described in this section.

12.2 Registers

The MCU contains five registers which are hard-wired within the CPU and are not part of the memory map. These five registers are shown in Figure 12-1 and are described in the following paragraphs.

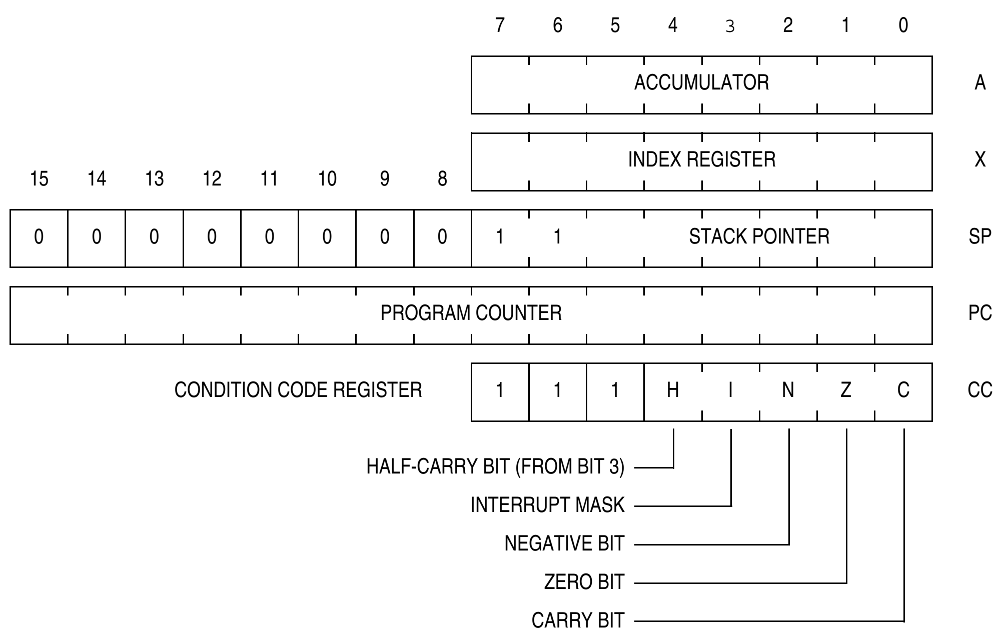


Figure 12-1. MC68HC05 Programming Model

12.2.1 Accumulator

The accumulator is a general-purpose 8-bit register as shown in Figure 12-1. The CPU uses the accumulator to hold operands and results of arithmetic calculations or non-arithmetic operations. The accumulator is unaffected by a reset of the device.

12.2.2 Index Register

The index register shown in Figure 12-1 is an 8-bit register that can perform two functions:

- Indexed addressing
- Temporary storage

In indexed addressing with no offset, the index register contains the low byte of the operand address, and the high byte is assumed to be $00. In indexed addressing with an 8-bit offset, the CPU finds the operand address by adding the index register contents to an 8-bit immediate value. In indexed addressing with a 16-bit offset, the CPU finds the operand address by adding the index register contents to a 16-bit immediate value.

The index register can also serve as an auxiliary accumulator for temporary storage. The index register is unaffected by a reset of the device.

12.2.3 Stack Pointer

The stack pointer shown in Figure 12-1 is a 16-bit register internally. In devices with memory maps less than 64 Kbytes, the unimplemented upper address lines are ignored. The stack pointer contains the address of the next free location on the stack. During a reset or the reset stack pointer (RSP) instruction, the stack pointer is set to $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the 10 most significant bits are permanently set to 0000000011. The six least significant register bits are appended to these 10 fixed bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 ($40) locations. If 64 locations are exceeded, the stack pointer wraps around and writes over the previously stored information. A subroutine call occupies two locations on the stack and an interrupt uses five locations.

12.2.4 Program Counter

The program counter shown in Figure 12-1 is a 16-bit register internally. In devices with memory maps less than 64 Kbytes, the unimplemented upper address lines are ignored. The program counter contains the address of the next instruction or operand to be fetched.

Normally, the address in the program counter increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

12.2.5 Condition Code Register

The CCR shown in Figure 12-1 is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. The fifth bit is the interrupt mask. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. The condition code register should be thought of as having three additional upper bits that are always ones. Only the interrupt mask is affected by a reset of the device. The following paragraphs explain the functions of the lower five bits of the condition code register.

H — Half Carry Bit

When the half-carry bit is set, it means that a carry occurred between bits 3 and 4 of the accumulator during the last ADD or ADC (add with carry) operation. The half-carry bit is required for binary-coded decimal (BCD) arithmetic operations.

I — Interrupt Mask Bit

When the interrupt mask is set, the internal and external interrupts are disabled. Interrupts are enabled when the interrupt mask is cleared. When an interrupt occurs, the interrupt mask is automatically set after the CPU registers are saved on the stack, but before the interrupt vector is fetched. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. Normally, the interrupt is processed as soon as the interrupt mask is cleared.

A return from interrupt (RTI) instruction pulls the CPU registers from the stack, restoring the interrupt mask to its state before the interrupt was encountered. After any reset, the interrupt mask is set and can only be cleared by the clear I bit (CLI), STOP, or WAIT instructions.

N — Negative Bit

The negative bit is set when the result of the last arithmetic operation, logical operation, or data manipulation was negative. (Bit 7 of the result was a logic one.)

The negative bit can also be used to check an often-tested flag by assigning the flag to bit 7 of a register or memory location. Loading the accumulator with the contents of that register or location then sets or clears the negative bit according to the state of the flag.

Z — Zero Bit

The zero bit is set when the result of the last arithmetic operation, logical operation, data manipulation, or data load operation was zero.

C — Carry/Borrow Bit

The carry/borrow bit is set when a carry out of bit 7 of the accumulator occurred during the last arithmetic operation, logical operation, or data manipulation. The carry/borrow bit is also set or cleared during bit test and branch instructions and during shifts and rotates. This bit is not set by an INC or DEC instruction.

Chapter 13
Instruction Set

13.1 Introduction

The MCU instruction set has 62 instructions and uses eight addressing modes. The instructions include all those of the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. The MUL instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is stored in the index register, and the low-order product is stored in the accumulator.

13.2 Addressing Modes

The CPU uses eight addressing modes for flexibility in accessing data. The addressing modes provide eight different ways for the CPU to find the data required to execute an instruction. The eight addressing modes are:

- Inherent
- Immediate
- Direct
- Extended
- Indexed, no offset
- Indexed, 8-bit offset
- Indexed, 16-bit offset
- Relative

13.2.1 Inherent

Inherent instructions are those that have no operand, such as return from interrupt (RTI) and stop (STOP). Some of the inherent instructions act on data in the CPU registers, such as set carry flag (SEC) and increment accumulator (INCA). Inherent instructions require no operand address and are one byte long.

13.2.2 Immediate

Immediate instructions are those that contain a value to be used in an operation with the value in the accumulator or index register. Immediate instructions require no operand address and are two bytes long. The opcode is the first byte, and the immediate data value is the second byte.

13.2.3 Direct

Direct instructions can access any of the first 256 memory locations with two bytes. The first byte is the opcode, and the second is the low byte of the operand address. In direct addressing, the CPU automatically uses $00 as the high byte of the operand address.

13.2.4 Extended

Extended instructions use three bytes and can access any address in memory. The first byte is the opcode; the second and third bytes are the high and low bytes of the operand address.

When using the Freescale assembler, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.

13.2.5 Indexed, No Offset

Indexed instructions with no offset are 1-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the effective address of the operand. The CPU automatically uses $00 as the high byte, so these instructions can address locations $0000–$00FF.

Indexed, no offset instructions are often used to move a pointer through a table or to hold the address of a frequently used RAM or I/O location.

13.2.6 Indexed, 8-Bit Offset

Indexed, 8-bit offset instructions are 2-byte instructions that can access data with variable addresses within the first 511 memory locations. The CPU adds the unsigned byte in the index register to the unsigned byte following the opcode. The sum is the effective address of the operand. These instructions can access locations $0000–$01FE.

Indexed 8-bit offset instructions are useful for selecting the kth element in an n-element table. The table can begin anywhere within the first 256 memory locations and could extend as far as location 510 ($01FE). The k value is typically in the index register, and the address of the beginning of the table is in the byte following the opcode.

13.2.7 Indexed,16-Bit Offset

Indexed, 16-bit offset instructions are 3-byte instructions that can access data with variable addresses at any location in memory. The CPU adds the unsigned byte in the index register to the two unsigned bytes following the opcode. The sum is the effective address of the operand. The first byte after the opcode is the high byte of the 16-bit offset; the second byte is the low byte of the offset.

Indexed, 16-bit offset instructions are useful for selecting the kth element in an n-element table anywhere in memory.

As with direct and extended addressing, the Freescale assembler determines the shortest form of indexed addressing.

13.2.8 Relative

Relative addressing is only for branch instructions. If the branch condition is true, the CPU finds the effective branch destination by adding the signed byte following the opcode to the contents of the program counter. If the branch condition is not true, the CPU goes to the next instruction. The offset is a signed, two's complement byte that gives a branching range of –128 to +127 bytes from the address of the next location after the branch instruction.

When using the Freescale assembler, the programmer does not need to calculate the offset, because the assembler determines the proper offset and verifies that it is within the span of the branch.

13.3 Instruction Types

The MCU instructions fall into the following five categories:

- Register/memory instructions
- Read-modify-write instructions
- Jump/branch instructions
- Bit manipulation instructions
- Control instructions

13.3.1 Register/Memory Instructions

These instructions operate on CPU registers and memory locations. Most of them use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory.

Table 13-1. Register/Memory Instructions

Instruction	Mnemonic
Add Memory Byte and Carry Bit to Accumulator	ADC
Add Memory Byte to Accumulator	ADD
AND Memory Byte with Accumulator	AND
Bit Test Accumulator	BIT
Compare Accumulator	CMP
Compare Index Register with Memory Byte	CPX
EXCLUSIVE OR Accumulator with Memory Byte	EOR
Load Accumulator with Memory Byte	LDA
Load Index Register with Memory Byte	LDX
Multiply	MUL
OR Accumulator with Memory Byte	ORA
Subtract Memory Byte and Carry Bit from Accumulator	SBC
Store Accumulator in Memory	STA
Store Index Register in Memory	STX
Subtract Memory Byte from Accumulator	SUB

13.3.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to the memory location or to the register.

NOTE
Do not use read modify-write operations on write-only registers.

Table 13-2. Read-Modify-Write Instructions

Instruction	Mnemonic
Arithmetic Shift Left (Same as LSL)	ASL
Arithmetic Shift Right	ASR
Bit Clear	BCLR(1)
Bit Set	BSET(1)
Clear Register	CLR
Complement (One's Complement)	COM
Decrement	DEC
Increment	INC
Logical Shift Left (Same as ASL)	LSL
Logical Shift Right	LSR
Negate (Two's Complement)	NEG
Rotate Left through Carry Bit	ROL
Rotate Right through Carry Bit	ROR
Test for Negative or Zero	TST(2)

1. Unlike other read-modify-write instructions, BCLR and BSET use only direct addressing.
2. TST is an exception to the read-modify-write sequence because it does not write a replacement value.

13.3.3 Jump/Branch Instructions

Jump instructions allow the CPU to interrupt the normal sequence of the program counter. The unconditional jump instruction (JMP) and the jump-to-subroutine instruction (JSR) have no register operand. Branch instructions allow the CPU to interrupt the normal sequence of the program counter when a test condition is met. If the test condition is not met, the branch is not performed.

The BRCLR and BRSET instructions cause a branch based on the state of any readable bit in the first 256 memory locations. These 3-byte instructions use a combination of direct addressing and relative addressing. The direct address of the byte to be tested is in the byte following the opcode. The third byte is the signed offset byte. The CPU finds the effective branch destination by adding the third byte to the program counter if the specified bit tests true. The bit to be tested and its condition (set or clear) is part of the opcode. The span of branching is from –128 to +127 from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register.

Table 13-3. Jump and Branch Instructions

Instruction	Mnemonic
Branch if Carry Bit Clear	BCC
Branch if Carry Bit Set	BCS
Branch if Equal	BEQ
Branch if Half-Carry Bit Clear	BHCC
Branch if Half-Carry Bit Set	BHCS
Branch if Higher	BHI
Branch if Higher or Same	BHS
Branch if $\overline{\text{IRQ}}$ Pin High	BIH
Branch if $\overline{\text{IRQ}}$ Pin Low	BIL
Branch if Lower	BLO
Branch if Lower or Same	BLS
Branch if Interrupt Mask Clear	BMC
Branch if Minus	BMI
Branch if Interrupt Mask Set	BMS
Branch if Not Equal	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit Clear	BRCLR
Branch Never	BRN
Branch if Bit Set	BRSET
Branch to Subroutine	BSR
Unconditional Jump	JMP
Jump to Subroutine	JSR

13.3.4 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory, which includes I/O registers and on-chip RAM locations. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations.

Table 13-4. Bit Manipulation Instructions

Instruction	Mnemonic
Bit Clear	BCLR
Branch if Bit Clear	BRCLR
Branch if Bit Set	BRSET
Bit Set	BSET

13.3.5 Control Instructions

These instructions act on CPU registers and control CPU operation during program execution.

Table 13-5. Control Instructions

Instruction	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
No Operation	NOP
Reset Stack Pointer	RSP
Return from Interrupt	RTI
Return from Subroutine	RTS
Set Carry Bit	SEC
Set Interrupt Mask	SEI
Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin	STOP
Software Interrupt	SWI
Transfer Accumulator to Index Register	TAX
Transfer Index Register to Accumulator	TXA
Stop CPU Clock and Enable Interrupts	WAIT

13.4 Instruction Set Summary

Table 13-6 is an alphabetical list of all M68HC05 instructions and shows the effect of each instruction on the condition code register.

Table 13-6. Instruction Set Summary (Sheet 1 of 6)

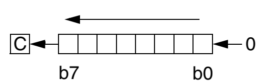
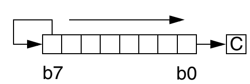
Source Form	Operation	Description	Effect on CCR H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X	Add without Carry	A ← (A) + (M)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	AB BB CB DB EB FB	ii dd hh ll ee ff ff	2 3 4 5 4 3
AND #*opr* AND *opr* AND *opr* AND *opr*,X AND *opr*,X AND ,X	Logical AND	A ← (A) ∧ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A4 B4 C4 D4 E4 F4	ii dd hh ll ee ff ff	2 3 4 5 4 3
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X	Arithmetic Shift Left (Same as LSL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR ,X	Arithmetic Shift Right		—	—	↕	↕	↕	DIR INH INH IX1 IX	37 47 57 67 77	dd ff 	5 3 3 6 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BCLR *n opr*	Clear Bit n	Mn ← 0	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? Z = 1	—	—	—	—	—	REL	27	rr	3
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3

Table 13-6. Instruction Set Summary (Sheet 2 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BIH *rel*	Branch if IRQ Pin High	PC ← (PC) + 2 + *rel* ? IRQ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if IRQ Pin Low	PC ← (PC) + 2 + *rel* ? IRQ = 0	—	—	—	—	—	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X	Bit Test Accumulator with Memory Byte	(A) ∧ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A5 B5 C5 D5 E5 F5	ii dd hh ll ee ff ff	2 3 4 5 4 3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? C ∨ Z = 1	—	—	—	—	—	REL	23	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? I = 0	—	—	—	—	—	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? N = 1	—	—	—	—	—	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? I = 1	—	—	—	—	—	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? Z = 0	—	—	—	—	—	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? N = 0	—	—	—	—	—	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel* ? 1 = 1	—	—	—	—	—	REL	20	rr	3
BRCLR *n opr rel*	Branch if Bit n Clear	PC ← (PC) + 2 + *rel* ? Mn = 0	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3
BRSET *n opr rel*	Branch if Bit n Set	PC ← (PC) + 2 + *rel* ? Mn = 1	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BSET *n opr*	Set Bit n	Mn ← 1	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	—	—	—	—	—	REL	AD	rr	6
CLC	Clear Carry Bit	C ← 0	—	—	—	—	0	INH	98		2
CLI	Clear Interrupt Mask	I ← 0	—	0	—	—	—	INH	9A		2

Table 13-6. Instruction Set Summary (Sheet 3 of 6)

Source Form	Operation	Description	H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
CLR *opr*	Clear Byte	M ← $00	—	—	0	1	—	DIR	3F	dd	5
CLRA		A ← $00						INH	4F		3
CLRX		X ← $00						INH	5F		3
CLR *opr*,X		M ← $00						IX1	6F	ff	6
CLR ,X		M ← $00						IX	7F		5
CMP #*opr*	Compare Accumulator with Memory Byte	(A) – (M)	—	—	↕	↕	↕	IMM	A1	ii	2
CMP *opr*								DIR	B1	dd	3
CMP *opr*								EXT	C1	hh ll	4
CMP *opr*,X								IX2	D1	ee ff	5
CMP *opr*,X								IX1	E1	ff	4
CMP ,X								IX	F1		3
COM *opr*	Complement Byte (One's Complement)	M ← ($\overline{M}$) = $FF – (M)	—	—	↕	↕	1	DIR	33	dd	5
COMA		A ← ($\overline{A}$) = $FF – (A)						INH	43		3
COMX		X ← ($\overline{X}$) = $FF – (X)						INH	53		3
COM *opr*,X		M ← ($\overline{M}$) = $FF – (M)						IX1	63	ff	6
COM ,X		M ← ($\overline{M}$) = $FF – (M)						IX	73		5
CPX #*opr*	Compare Index Register with Memory Byte	(X) – (M)	—	—	↕	↕	↕	IMM	A3	ii	2
CPX *opr*								DIR	B3	dd	3
CPX *opr*								EXT	C3	hh ll	4
CPX *opr*,X								IX2	D3	ee ff	5
CPX *opr*,X								IX1	E3	ff	4
CPX ,X								IX	F3		3
DEC *opr*	Decrement Byte	M ← (M) – 1	—	—	↕	↕	—	DIR	3A	dd	5
DECA		A ← (A) – 1						INH	4A		3
DECX		X ← (X) – 1						INH	5A		3
DEC *opr*,X		M ← (M) – 1						IX1	6A	ff	6
DEC ,X		M ← (M) – 1						IX	7A		5
EOR #*opr*	EXCLUSIVE OR Accumulator with Memory Byte	A ← (A) ⊕ (M)	—	—	↕	↕	—	IMM	A8	ii	2
EOR *opr*								DIR	B8	dd	3
EOR *opr*								EXT	C8	hh ll	4
EOR *opr*,X								IX2	D8	ee ff	5
EOR *opr*,X								IX1	E8	ff	4
EOR ,X								IX	F8		3
INC *opr*	Increment Byte	M ← (M) + 1	—	—	↕	↕	—	DIR	3C	dd	5
INCA		A ← (A) + 1						INH	4C		3
INCX		X ← (X) + 1						INH	5C		3
INC *opr*,X		M ← (M) + 1						IX1	6C	ff	6
INC ,X		M ← (M) + 1						IX	7C		5
JMP *opr*	Unconditional Jump	PC ← Jump Address	—	—	—	—	—	DIR	BC	dd	2
JMP *opr*								EXT	CC	hh ll	3
JMP *opr*,X								IX2	DC	ee ff	4
JMP *opr*,X								IX1	EC	ff	3
JMP ,X								IX	FC		2
JSR *opr*	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Effective Address	—	—	—	—	—	DIR	BD	dd	5
JSR *opr*								EXT	CD	hh ll	6
JSR *opr*,X								IX2	DD	ee ff	7
JSR *opr*,X								IX1	ED	ff	6
JSR ,X								IX	FD		5

Table 13-6. Instruction Set Summary (Sheet 4 of 6)

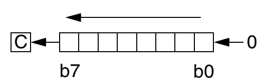
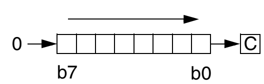
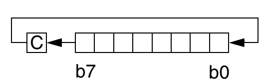
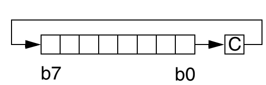
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X	Load Accumulator with Memory Byte	A ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff	2 3 4 5 4 3
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X	Load Index Register with Memory Byte	X ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff	2 3 4 5 4 3
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X	Logical Shift Left (Same as ASL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff	5 3 3 6 5
LSR *opr* LSRA LSRX LSR *opr*,X LSR ,X	Logical Shift Right		—	—	0	↕	↕	DIR INH INH IX1 IX	34 44 54 64 74	dd ff	5 3 3 6 5
MUL	Unsigned Multiply	X : A ← (X) × (A)	0	—	—	—	0	INH	42		1 1
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	—	—	↕	↕	↕	DIR INH INH IX1 IX	30 40 50 60 70	dd ff	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	39 49 59 69 79	dd ff	5 3 3 6 5
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	36 46 56 66 76	dd ff	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2

Table 13-6. Instruction Set Summary (Sheet 5 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	INH	80		9
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	—	—	—	—	—	INH	81		6
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X	Store Accumulator in Memory	M ← (A)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff	4 5 6 5 4
STOP	Stop Oscillator and Enable IRQ Pin		—	0	—	—	—	INH	8E		2
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X	Store Index Register In Memory	M ← (X)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff	4 5 6 5 4
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X	Subtract Memory Byte from Accumulator	A ← (A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff	2 3 4 5 4 3
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		10
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2
TST *opr* TSTA TSTX TST *opr*,X TST ,X	Test Memory Byte for Negative or Zero	(M) – $00	—	—	↕	↕	—	DIR INH INH IX1 IX	3D 4D 5D 6D 7D	dd ff	4 3 3 5 4

Table 13-6. Instruction Set Summary (Sheet 6 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
TXA	Transfer Index Register to Accumulator	A ← (X)	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	0	—	—	—	INH	8F		2

A	Accumulator	*opr*	Operand (one or two bytes)
C	Carry/borrow flag	PC	Program counter
CCR	Condition code register	PCH	Program counter high byte
dd	Direct address of operand	PCL	Program counter low byte
dd rr	Direct address of operand and relative offset of branch instruction	REL	Relative addressing mode
DIR	Direct addressing mode	*rel*	Relative program counter offset byte
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	rr	Relative program counter offset byte
EXT	Extended addressing mode	SP	Stack pointer
ff	Offset byte in indexed, 8-bit offset addressing	X	Index register
H	Half-carry flag	Z	Zero flag
hh ll	High and low bytes of operand address in extended addressing	#	Immediate value
I	Interrupt mask	∧	Logical AND
ii	Immediate operand byte	∨	Logical OR
IMM	Immediate addressing mode	⊕	Logical EXCLUSIVE OR
INH	Inherent addressing mode	( )	Contents of
IX	Indexed, no offset addressing mode	–( )	Negation (two's complement)
IX1	Indexed, 8-bit offset addressing mode	←	Loaded with
IX2	Indexed, 16-bit offset addressing mode	?	If
M	Memory location	:	Concatenated with
N	Negative flag	↕	Set or cleared
n	Any bit	—	Not affected

13.5 Opcode Map

See Table 13-7.

Table 13-7. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	DIR	DIR	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
MSB / LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	MSB / LSB
0	5 BRSET0 3 DIR	5 BSET0 2 DIR	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0
1	5 BRCLR0 3 DIR	5 BCLR0 2 DIR	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	1
2	5 BRSET1 3 DIR	5 BSET1 2 DIR	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	2
3	5 BRCLR1 3 DIR	5 BCLR1 2 DIR	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	3
4	5 BRSET2 3 DIR	5 BSET2 2 DIR	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	4
5	5 BRCLR2 3 DIR	5 BCLR2 2 DIR	3 BCS/BLO 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	5
6	5 BRSET3 3 DIR	5 BSET3 2 DIR	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	6
7	5 BRCLR3 3 DIR	5 BCLR3 2 DIR	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	7
8	5 BRSET4 3 DIR	5 BSET4 2 DIR	3 BHCC 2 REL	5 ASL/LSL 2 DIR	3 ASLA/LSLA 1 INH	3 ASLX/LSLX 1 INH	6 ASL/LSL 2 IX1	5 ASL/LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	8
9	5 BRCLR4 3 DIR	5 BCLR4 2 DIR	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9
A	5 BRSET5 3 DIR	5 BSET5 2 DIR	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A
B	5 BRCLR5 3 DIR	5 BCLR5 2 DIR	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	B
C	5 BRSET6 3 DIR	5 BSET6 2 DIR	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C
D	5 BRCLR6 3 DIR	5 BCLR6 2 DIR	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D
E	5 BRSET7 3 DIR	5 BSET7 2 DIR	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	E
F	5 BRCLR7 3 DIR	5 BCLR7 2 DIR	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F

INH = Inherent
IMM = Immediate
DIR = Direct
EXT = Extended

REL = Relative
IX = Indexed, No Offset
IX1 = Indexed, 8-Bit Offset
IX2 = Indexed, 16-Bit Offset

MSB / LSB	0	MSB of Opcode in Hexadecimal
0	5 BRSET0 3 DIR	5 = Number of Cycles; BRSET0 = Opcode Mnemonic; 3 DIR = Number of Bytes/Addressing Mode

LSB of Opcode in Hexadecimal

Chapter 14
Electrical Specifications

14.1 Introduction

This section contains the electrical and timing specifications.

14.2 Maximum Ratings

Maximum ratings are the extreme limits to which the MCU can be exposed without permanently damaging it.

The MCU contains circuitry to protect the inputs against damage from high static voltages; however, do not apply voltages higher than those shown in the table below. Keep V_{In} and V_{Out} within the range $V_{SS} \leq (V_{In}$ or $V_{Out}) \leq V_{DD}$. Connect unused inputs to the appropriate voltage level, either V_{SS} or V_{DD}.

Rating[(1)]	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to +7.0	V
Input voltage	V_{In}	V_{SS} –0.3 to V_{DD} +0.3	V
Bootloader mode ($\overline{IRQ}/V_{PP}$ pin only)	V_{In}	V_{SS} –0.3 to 2 x V_{DD} +0.3	V
Current drain per pin excluding V_{DD} and V_{SS}	I	25	mA
Storage temperature range	T_{stg}	–65 to +150	°C

1. Voltages are referenced to V_{SS}.

NOTE

This device is not guaranteed to operate properly at the maximum ratings. Refer to 14.5 5.0-Volt DC Electrical Characteristics and 14.6 3.3-Volt DC Electrical Charactertistics for guaranteed operating conditions.

14.3 Operating Temperature Range

Characteristic	Symbol	Value	Unit
Operating temperature range MC68HC705P6A (standard) MC68HC705P6AC (extended)	T_A	T_L to T_H 0 to +70 –40 to +85	°C

14.4 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance PDIP SOIC	θ_{JA}	 60 60	°C/W

14.5 5.0-Volt DC Electrical Characteristics

Characteristic(1)	Symbol	Min	Typ(2)	Max	Unit
Output voltage I_{Load} = 10.0 µA I_{Load} = –10.0 µA	V_{OL} V_{OH}	— V_{DD} –0.1	— —	0.1 —	V
Output high voltage (I_{Load} = –0.8 mA) PA0:7, PB5:7, PC2:7, PD5, TCMP (I_{Load} = –5.0 mA) PC0:1	V_{OH}	V_{DD} –0.8 V_{DD} –0.8	— —	— —	V
Output low voltage (I_{Load} = 1.6 mA) PA0:7, PB5:7, PC2:7, PD5, TCMP (I_{Load} = 10 mA) PC0:1	V_{OL}	— —	— —	0.4 0.4	V
Input high voltage PA0:7, PB5:7, PC0:7, PD5, TCAP/PD7, $\overline{IRQ}/V_{PP}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input low voltage PA0:7, PB5:7, PC0:7, PD5, TCAP/PD7, $\overline{IRQ}/V_{PP}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Supply current(3), (4) Run Wait(5) (A/D converter on) Wait(5) (A/D converter off) Stop(6) 25°C 0°C to +70°C (standard) –40°C to +85°C (extended)	I_{DD}	 — — — — — —	 4.0 2.0 1.3 2 — —	 7.0 4.0 2.0 30 50 100	 mA mA mA µA µA µA
I/O ports high-z leakage current PA0:7, PB5:7, PC0:7, PD5, TCAP/PD7	I_{IL}	—	—	±10.0	µA
A/D ports hi-z leakage current PC3:7	I_{OZ}	—	—	±1.0	µA
Input current $\overline{RESET}$, $\overline{IRQ}/V_{PP}$, OSC1, PD7/TCAP	I_{In}	—	—	±1.0	µA
Input pullup current PA0:7 (with pullup enabled)	I_{In}	175	385	750	µA
Capaitance Ports (as input or output) $\overline{RESET}$, $\overline{IRQ}/V_{PP}$	C_{Out} C_{In}	— —	— —	12 8	pF

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = –40°C to +85°C, unless otherwise noted. All values shown refelect pre-silicon estimates.
2. Typical values at midpoint of voltage range, 25°C only.
3. Run (Operating) I_{DD}, Wait I_{DD}: To be measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
4. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V.
5. Wait I_{DD} will be affected linearly by the OSC2 capacitance.
6. Stop I_{DD} to be measured with OSC1 = V_{SS}.

14.6 3.3-Volt DC Electrical Charactertistics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output voltage I_{Load} = 10.0 μA I_{Load} = –10.0 μA	V_{OL} V_{OH}	— V_{DD} –0.1	— —	0.1 —	V
Output high voltage (I_{Load} = –0.2 mA) PA0:7, PB5:7, PC2:7, PD5, TCMP (I_{Load} = –1.2 mA) PC0:1	V_{OH}	V_{DD} –0.3 V_{DD} –0.3	— —	— —	V
Output low voltage (I_{Load} = 0.4 mA) PA0:7, PB5:7, PC2:7, PD5, TCMP (I_{Load} = 2.5 mA) PC0:1	V_{OL}	— —	— —	0.3 0.3	V
Input high voltage PA0:7, PB5:7, PC0:7, PD5, TCAP/PD7, $\overline{IRQ}/V_{PP}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input low voltage PA0:7, PB5:7, PC0:7, PD5, TCAP/PD7, $\overline{IRQ}/V_{PP}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Supply current[3], [4] Run Wait[5] (A/D converter on) Wait[5] (A/D converter off) Stop[6] 25°C 0°C to +70°C (standard) –40°C to +85°C (extended)	I_{DD}	 — — — — — —	 1.8 1.0 0.6 2 — —	 2.5 1.4 1.0 20 40 50	 mA mA mA μA μA μA
I/O ports high-z leakage current PA0:7, PB5:7, PC0:7, PD5, TCAP/PD7	I_{IL}	—	—	±10.0	μA
A/D ports hi-z leakage current PC3:7	I_{OZ}	—	—	±1.0	μA
Input current $\overline{RESET}$, $\overline{IRQ}/V_{PP}$, OSC1, PD7/TCAP	I_{In}	—	—	±1.0	μA
Input pullup current PA0:7 (with pullup enabled)	I_{In}	75	175	350	μA
Capaitance Ports (as input or output) $\overline{RESET}$, $\overline{IRQ}/V_{PP}$	C_{Out} C_{In}	— —	— —	12 8	pF

1. V_{DD} = 3.3 Vdc ± 0.3 Vdc, V_{SS} = 0 Vdc, T_A = –40°C to +85°C, unless otherwise noted. All values shown reflect pre-silicon estimates.
2. Typical values at midpoint of voltage range, 25°C only.
3. Run (Operating) I_{DD}, Wait I_{DD}: To be measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
4. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V.
5. Wait I_{DD} will be affected linearly by the OSC2 capacitance.
6. Stop I_{DD} to be measured with OSC1 = V_{SS}.

14.7 A/D Converter Characteristics

Characteristic[1]	Min	Max	Unit	Comments
Resolution	8	8	Bits	
Absolute accuacy ($V_{DD} \geq V_{REFH} > 4.0$)	—	± 1 1/2	LSB	Including quanitization
Conversion range V_{REFH}	V_{SS} V_{SS}	V_{REFH} V_{DD}	V	A/D accuracy may decrease proportionately as V_{REFH} is reduced below 4.0
Input leakage AD0, AD1, AD2, AD3 V_{REFH}	— —	± 1 ± 1	µA	
Conversion time MCU external oscillator Internal RC oscillator	— —	32 32	t_{cyc} µs	Includes sampling time
Monotonicity	Inherent (within total error)			
Zero input reading	00	01	Hex	V_{in} = 0 V
Full-scale reading	FE	FF	Hex	$V_{in} = V_{REFH}$
Sample time MCU external oscillator Internal RC oscillator	— —	12 12	t_{cyc} µs	
Input capacitance	—	12	pF	
Analog input voltage	V_{SS}	V_{REFH}	V	
A/D on current stabilization time	—	100	µs	t_{ADON}
A/D ports hi-z leakage current (PC3:7)	—	± 1	µA	I_{OZ}

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = –40°C to +85°C, unless otherwise noted.

14.8 EPROM Programming Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Programming voltage $\overline{IRQ}/V_{PP}$	V_{PP}	16.25	16.5	16.75	V
Programming current $\overline{IRQ}/V_{PP}$	I_{PP}	—	5.0	10	mA
Programming time per byte	t_{EPGM}	4	—	—	ms

14.9 SIOP Timing

Number	Characteristic	Symbol	Min	Max	Unit
	Operating frequency Master Slave	$f_{op(m)}$ $f_{op(s)}$	0.25 dc	0.25 0.25	f_{op}
1	Cycle time Master Slave	$t_{cyc(m)}$ $t_{cyc(s)}$	4.0 —	4.0 4.0	t_{cyc}
2	SCK low time	t_{cyc}	932	—	ns
3	SDO data valid time	t_v	—	200	ns
4	SDO hold time	t_{ho}	0	—	ns
5	SDI setup time	t_s	100	—	ns
6	SDI hold time	t_h	100	—	ns

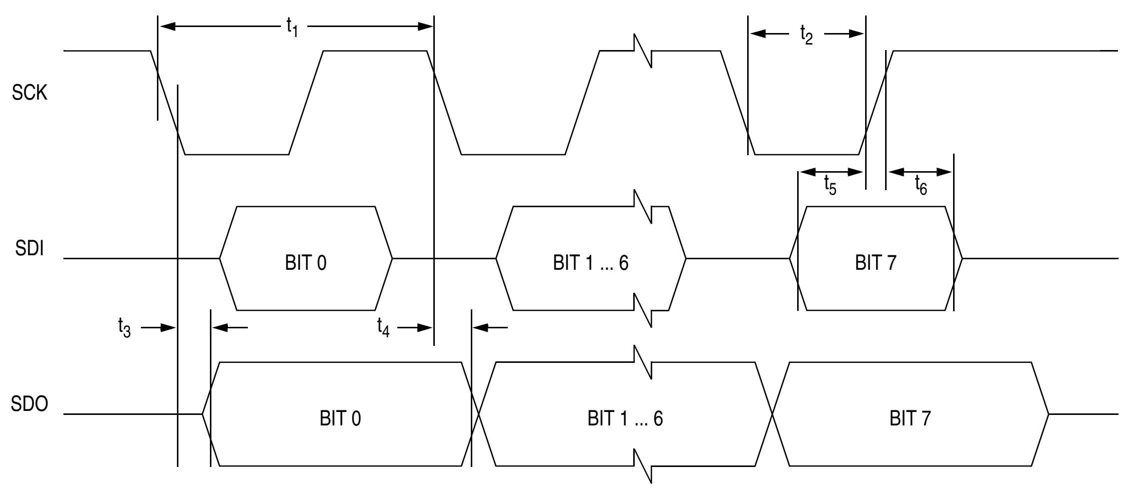
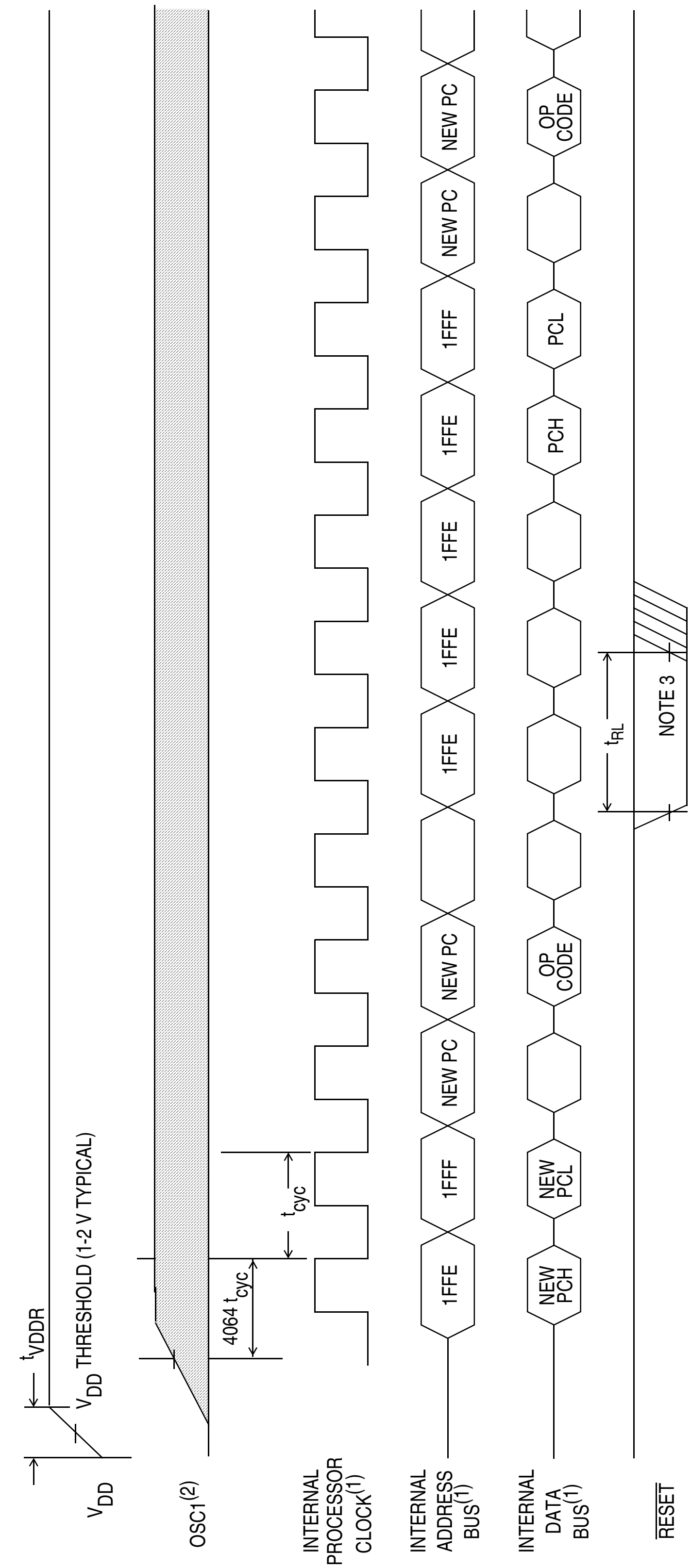


Figure 14-1. SIOP Timing Diagram

14.10 Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Frequency of operation Crystal option External clock option	f_{OSC}	— DC	4.2 4.2	MHz
Internal operating frequency Crystal ($f_{OSC} \div 2$) External clock ($f_{OSC} \div 2$)	f_{OP}	— DC	2.1 2.1	MHz
Cycle time	t_{CYC}	476	—	ns
Crystal oscillator startup time	t_{OXOV}	—	100	ms
Stop mode recovery startup time (crystal oscillator)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ pulse width	t_{RL}	1.5	—	t_{CYC}
Interrupt pulse width low (edge-triggered)	t_{ILIH}	125	—	ns
Interrupt pulse period[2]	t_{ILIL}	Note 2	—	t_{CYC}
OSC1 pulse width	t_{OH}, t_{OL}	200	—	ns
A/D On current stabilization time	t_{ADON}	Q	100	µs

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = –40°C to +125°C, unless otherwise noted
2. The minimum period, t_{ILIL}, should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{CYC}.



Notes:
1. Internal timing signal and bus information are not available externally.
2. OSC1 line is not meant to represent frequency. It is only used to represent time.
3. The next rising edge of the internal clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.

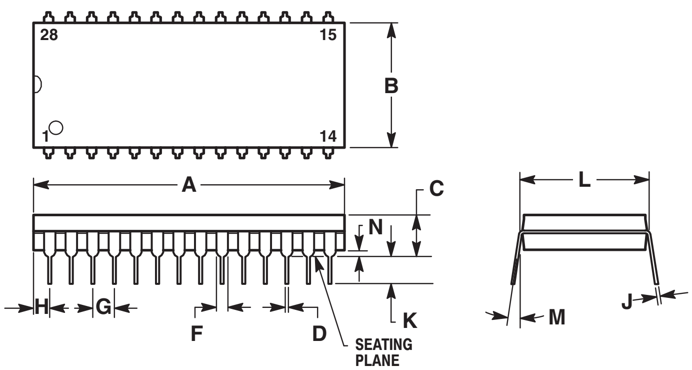
Figure 14-2. Power-On Reset and External Reset Timing Diagram

Chapter 15
Mechanical Specifications

15.1 Introduction

The MC68HC705P6A is available in either a 28-pin plastic dual in-line (PDIP) or a 28-pin small outline integrated circuit (SOIC) package.

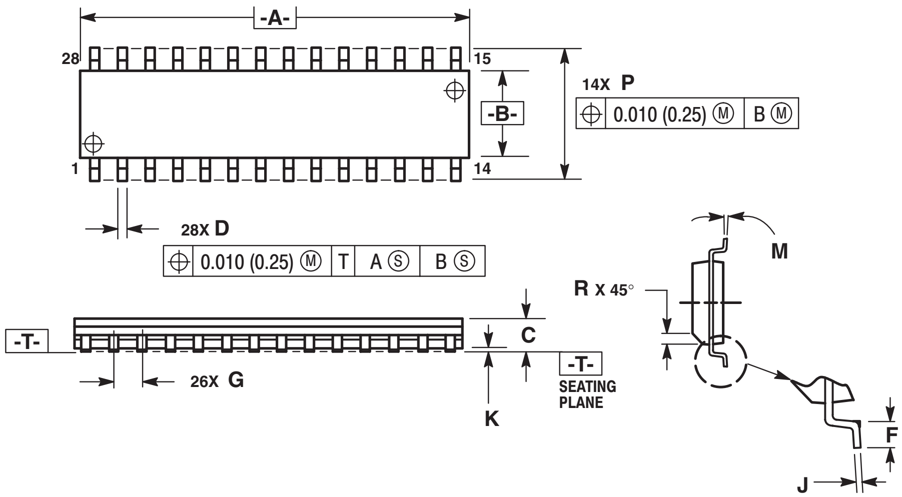
15.2 Plastic Dual In-Line Package (Case 710)



NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

15.3 Small Outline Integrated Circuit Package (Case 751F)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 (0.005) TOTAL IN EXCESS OF D DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	17.80	18.05	0.701	0.711
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.41	0.90	0.016	0.035
G	1.27 BSC		0.050 BSC	
J	0.23	0.32	0.009	0.013
K	0.13	0.29	0.005	0.011
M	0°	8°	0°	8°
P	10.05	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

Chapter 16
Ordering Information

16.1 Introduction

This section contains ordering information for the available package types.

16.2 MC Order Numbers

The following table shows the MC order numbers for the available package types.

MC Order Number	Operating Temperature Range
MC68HC705P6ACP[1] (extended)	–40°C to 85°C
MC68HC705P6ACDW[2] (extended)	–40°C to 85°C

1. P = Plastic dual in-line package
2. DW = Small outline integrated circuit (SOIC) package

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

MC68HC705P6A
Rev. 2.1, 9/2005

freescale™
semiconductor